

Tomoya Enokido Lu Yan  
Bin Xiao Daeyoung Kim  
Yuanshun Dai Laurence T. Yang (Eds.)

LNCS 3823

# Embedded and Ubiquitous Computing – EUC 2005 Workshops

**EUC 2005 Workshops:**  
UISW, NCUS, SecUbiq, USN, and TAUES  
Nagasaki, Japan, December 2005, Proceedings



ifip

 Springer

*Commenced Publication in 1973*

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

## Editorial Board

David Hutchison

*Lancaster University, UK*

Takeo Kanade

*Carnegie Mellon University, Pittsburgh, PA, USA*

Josef Kittler

*University of Surrey, Guildford, UK*

Jon M. Kleinberg

*Cornell University, Ithaca, NY, USA*

Friedemann Mattern

*ETH Zurich, Switzerland*

John C. Mitchell

*Stanford University, CA, USA*

Moni Naor

*Weizmann Institute of Science, Rehovot, Israel*

Oscar Nierstrasz

*University of Bern, Switzerland*

C. Pandu Rangan

*Indian Institute of Technology, Madras, India*

Bernhard Steffen

*University of Dortmund, Germany*

Madhu Sudan

*Massachusetts Institute of Technology, MA, USA*

Demetri Terzopoulos

*New York University, NY, USA*

Doug Tygar

*University of California, Berkeley, CA, USA*

Moshe Y. Vardi

*Rice University, Houston, TX, USA*

Gerhard Weikum

*Max-Planck Institute of Computer Science, Saarbruecken, Germany*

Tomoya Enokido Lu Yan Bin Xiao  
Daeyoung Kim Yuanshun Dai  
Laurence T. Yang (Eds.)

# Embedded and Ubiquitous Computing – EUC 2005 Workshops

EUC 2005 Workshops:  
UISW, NCUS, SecUbiq, USN, and TAUES  
Nagasaki, Japan, December 6-9, 2005  
Proceedings

## Volume Editors

Tomoya Enokido

Rissho University, Faculty of Business Administration

2-16 Osaki 4 Chome, Shinagawa-ku, Tokyo 141-8602, Japan

E-mail: eno@ris.ac.jp

Lu Yan

Turku Centre for Computer Science (TUUS)

Lemminkaisenkatu 14, 20520 Turku, Finland

E-mail: lyan@abo.fi

Bin Xiao

Hong Kong Polytechnic University, Department of Computing

Hung Hom, Kowloon, Hong Kong

E-mail: csbxiao@comp.polyu.edu.hk

Daeyoung Kim

Information and Communications University

119 Munji-ro, Yuseong-gu, Daejeon, 305-732, Korea

E-mail: kimd@icu.ac.kr

Yuanshun Dai

Purdue University, Department of Computer and Information Science

723 W. Michigan Street SL280, Indianapolis, IN 46202, USA

E-mail: ydai@cs.iupui.edu

Laurence T. Yang

St. Francis Xavier University, Department of Computer Science

Antigonish, NS, B2G 2W5, Canada

E-mail: lyang@stfx.ca

Library of Congress Control Number: 2005936805

CR Subject Classification (1998): C.2, C.3, D.4, D.2, H.4, K.6.5, H.5.3, K.4

ISSN 0302-9743

ISBN-10 3-540-30803-2 Springer Berlin Heidelberg New York

ISBN-13 978-3-540-30803-4 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media

springeronline.com

© IFIP International Federation for Information Processing 2005

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper SPIN: 11596042 06/3142 5 4 3 2 1 0

# Preface

Welcome to the proceedings of the EUC 2005 workshops, held in conjunction with the IFIP International Conference on Embedded and Ubiquitous Computing in Nagasaki, Japan, December 6-9, 2005.

The objective of these workshops is to extend the spectrum of the main conference by providing a premier international forum for researchers and practitioners from both industry and academia, to discuss hot topics and emerging areas, to share recent progress and latest results, and to promote cutting-edge research and future cooperation on embedded and ubiquitous computing.

To meet this objective, we featured five workshops:

- UISW 2005: The Second International Symposium on Ubiquitous Intelligence and Smart Worlds
- NCUS 2005: IFIP International Symposium on Network-Centric Ubiquitous Systems
- SecUbiq 2005: The First International Workshop on Security in Ubiquitous Computing Systems
- USN 2005: The First International Workshop on RFID and Ubiquitous Sensor Networks
- TAUES 2005: The International Workshop on Trusted and Autonomic Ubiquitous and Embedded Systems

They address five state-of-the-art research directions in embedded and ubiquitous computing:

- UISW 2005: Following ubiquitous computers, networks, information, services, etc., is a road towards a smart world (SW) created on both cyberspaces and real spaces. A SW is characterized mainly by ubiquitous intelligence (UI) or computational intelligence pervasive in the physical world, filled with ubiquitous intelligent or smart real things, that are capable of computing, communicating, and behaving smartly with some intelligence. One of the profound implications of such ubiquitous smart things is that various kinds and levels of intelligence will exist ubiquitously in everyday objects, environments, systems and even ourselves, and possibly be extended from man-made to natural things. Ubicomp or percomp can be regarded as the computing of all these intelligent/smart real things. A smart thing can be endowed with different levels of intelligence, and may be context-aware, active, interactive, reactive, proactive, assistive, adaptive, automated, sentient, perceptual, cognitive, autonomic and/or thinking. Intelligent/smart things is an emerging research field covering many disciplines. A series of grand challenges exist to move from the ubiquitous world with universal services of any means/place/time to the SW of trustworthy services with the right means/place/time.

- NCUS 2005: Historically, ubiquitous systems have been highly engineered for a particular task, with no spontaneous interactions among devices. Recent advances in wireless communication and sensor/actuator technologies have given rise to a new genre of ubiquitous systems. This new genre is characterized as self-organizing, critically resource constrained, and network-centric. The fundamental change is communication: numerous small devices operating collectively, rather than as stand-alone devices, form a dynamic ambient network that connects each device to more powerful networks and processing resources. IFIP International Symposium on Network-Centric Ubiquitous Systems was launched to serve as a premier international forum for researchers and practitioners, from both industry and academia to share the latest research results and ideas on ubiquitous networking and its applications, thereby promoting research activities in this area.
- SecUbiq 2005: Ubiquitous computing technology provides an environment where users expect to access resources and services anytime and anywhere. Serious security risks and problems arise because resources can now be accessed by almost anyone with a mobile device in such an open model. The security threats exploited the weakness of protocols as well as operating systems, and also extended to attack ubiquitous applications. The security issues, such as authentication, access control, trust management, privacy and anonymity etc., should be fully addressed. This workshop provided a forum for academic and industry professionals to discuss recent progress in the area of ubiquitous computing system security, and included studies on analysis, models and systems, new directions, and novel applications of established mechanisms approaching the risks and concerns associated with the utilization and acceptance of ubiquitous computing devices and systems.
- USN 2005: In the emerging era of ubiquitous computing, networked small embedded devices with sensing capabilities will play a key role. Small enough to guarantee the pervasiveness in the ubiquitous world, a network of sensor devices provides valuable information to be exploited for a great variety of sensor applications. While there has been intensive research during the last few years, the consideration of anywhere and anytime presence still brings new challenges, keeping the topic of sensor networks in the center of the ubiquitous systems investigation. At the same time, radio frequency identification (RFID) shows a great potential in market penetration to address today's object identification systems, and its technologies already entail a success for the industry with some field applications across the globe. However, numerous questions about its implementation, capability, performance, reliability, economy and integration with other technologies still remain to be answered. The purpose of USN 2005 was to establish a discussion framework on all the challenges raised from the evolution of the ubiquitous sensor networks and RFID technologies. As a unique opportunity to obtain an insight into the leading technologies of the next pervasive era, USN 2005 tried to provide the place for discussing and exchanging ideas from both academia and industry worldwide.

- TAUES 2005: Embedded and ubiquitous computing is emerging rapidly as an exciting new paradigm to provide computing and communication services all the time, everywhere. Its systems are now pervading every aspect of life to the point that they are hidden inside various appliances or can be worn unobtrusively as part of clothing and jewelry. To achieve this level of invisible ubiquitous and pervasive computation and communication, we will need to study trusted and self-managing infrastructure. As such, it is necessary to develop new trustworthy software, selfware technologies, and self-X properties to effectively and inconspicuously manage these emerging embedded and ubiquitous systems. Trustworthy computing, which is essential to embedded and ubiquitous systems, addresses all issues relating to security, privacy, reliability, and information integrity. One of the most promising paradigms for self-managing systems is that of autonomic computing, which is inspired by nature and biological systems (such as the autonomic nervous system) that have evolved to cope with the challenges of scale, complexity, heterogeneity and unpredictability by being decentralized, embedded, context aware, adaptive, ubiquitous and resilient. This new era is characterized by self-X properties such as self-defining, self-configuring, self-optimizing, self-protecting and self-healing as well as context aware and anticipatory. This workshop brought together computer scientists, industrial engineers and researchers to discuss and exchange experimental or theoretical results, novel designs, work-in-progress, experience, case studies, and trend-setting ideas in the area of trusted and autonomic ubiquitous and embedded systems.

In pursuit of excellence, a distinguished international panel of reviewers was assembled and worked hard to review the submitted papers in a timely and professional manner.

UISW 2005, NCUS 2005, SecUbiq 2005, and USN 2005 attracted 175, 66, 51, and 50 papers, respectively. The Program Committees accepted 59, 24, 21, and 18 papers based on peer reviews, for acceptance rates of 34%, 36%, 41%, and 36%, respectively. TAUES 2005 consists of ten accepted papers.

In total, 132 papers were chosen for delivery and inclusion in this volume from many submissions all over the world, with a weighted average acceptance rate of 36%. These, we believe, are of a high standard and resulted in stimulating discussions when presented at the forum.

Numerous people deserve appreciation and recognition for their contribution to making EUC 2005 workshops a success:

- UISW 2005: First of all, we would like to thank the EUC 2005 Organization Committee for their support, guidance, and help. We would like to express our special thanks to Jeneung Lee, Satoshi Itaya, Hiroyuki Yoshino, and Youhei Tanaka for maintaining the Web system, and handling the submission and review process. In addition, we would like to give our special thanks to local organizers at Nagasaki. Finally, we also would like to take the opportunity to thank all the members of the Organization Committee and Program Committee as well as the authors for paper submission and reviewers for paper review.

- NCUS 2005: The exciting program was the result of the hard and excellent work of many others. We would like to express our sincere appreciation to all authors for their valuable contributions and to all Program and Technical Committee members and external reviewers for their cooperation in completing the program under a very tight schedule.
- SecUbiq 2005: First, we would like to thank all the authors for their hard work in preparing submissions to the workshop. We deeply appreciate the effort and contributions of the Program Committee members, who worked very hard to send back their comments and to put together an exciting program. Especially, we thank the effort of those Program Committee members who delivered their reviews in a timely manner despite having to face very difficult personal situations. In addition, we would like to thank the EUC 2005 Organization Committee for their support, guidance, and help for the workshop. We would like to give our special thanks to the local organizers at Nagasaki Institute of Applied Science and to those people who kindly helped us prepare and organize the SecUbiq 2005 workshop.
- USN 2005: We owe a great deal of thanks to the members of the Program Committee and the reviewers. The success of this year's USN would not be possible without their hard work. We are also grateful to all the members of Steering Committee, Jongsuk Chae, Joongsoo Ma, Hao Min, Kang Shin and Yu-Chee Tseng, for their advice and support. Finally, our many thanks to Tomás Sánchez López of the Information and Communications University for his great help in preparing the workshop. USN 2005 was co-sponsored by the Mobile Multimedia Research Center (ITRC program of Ministry of Information and Communication, Korea) and the Auto-ID Labs Korea.
- TAUES 2005: The exciting program for this conference was the result of the hard and excellent work of many others, such as Program Co-chairs, external reviewers, Program and Technical Committee members. We would like to express our sincere appreciation to all authors for their valuable contributions and to all Program and Technical Committee members and external reviewers for their cooperation in completing the program under a very tight schedule. We were also grateful to the members of the Organizing Committee for supporting us in handling the many organizational tasks.

October 2005

Tomoya Enokido, Lu Yan  
Bin Xiao, Daeyoung Kim  
Yuanshun Dai, Laurence T. Yang  
EUC 2005 Workshop Chairs



# EUC 2005 Workshops Organization

## EUC 2005 Workshop Chairs

Makoto Takizawa           Tokyo Denki University, Japan  
Seongsoo Hong            Seoul National University, Korea

## UISW 2005 Executive Committee

General Chairs:            Jianhua Ma, Hosei University, Japan  
                              Laurence T. Yang, St. Francis Xavier University,  
                                  Canada

Program Chairs:           Tomoya Enokido, Rissho University, Japan  
                              Victor Callaghan, University of Essex, UK  
                              Hai Jin, Huazhong Univ. of Science & Technology,  
                                  China

Advisory Committee:      Makoto Takizawa, Tokyo Denki Univ., Japan  
                              Moon Hae Kim, Konkuk University, Korea  
                              Hitoshi Aida, The University of Tokyo, Japan  
                              Makoto Amamiya, Kyushu University, Japan  
                              Leonard Barolli, Fukuoka Institute of Tech., Japan  
                              Jingde Cheng, Saitama University, Japan  
                              Minyi Guo, The University of Aizu, Japan  
                              Ali R. Hurson, Pennsylvania State University, USA  
                              Haruhisa Ichikawa, NTT Network Innovation Lab.,  
                                  Japan  
                              Kane Kim, University of California, Irvine, USA  
                              Madjid Merabti, Liverpool John Moores Univ., UK  
                              Manish Parashar, Rutgers University, USA  
                              Tae-Woo Park, AFOSR/AOARD, USA  
                              Ichiro Satoh, National Institute of Informatics,  
                                  Japan  
                              Timothy K. Shih, Tamkang University, Taiwan  
                              David Taniar, Monash University, Australia  
                              Jeffrey J.P. Tsai, Univ. of Illinois at Chicago, USA  
                              Jhing-Fa Wang, Nat. Cheng Kung Univ., Taiwan  
                              Albert Zomaya, University of Sydney, Australia

## UISW 2005 Program/Technical Committee

Marios C. Angelides	Brunel University, UK
Bernady Apduhan	Kyushu Sangyo University, Japan
Juan Carlos Augusto	University of Ulster at Jordanstown, UK
Jiannong Cao	Hong Kong Polytechnic Univ., Hong Kong, China
Genci Capi	Fukuoka Institute of Tech., Japan
Chih-Yung Chang	Tamkang University, Taiwan
Han-Chieh Chao	National Dong Hwa University, Taiwan
Kuo-Ming Chao	Coventry University, UK
Vipin Chaudhary	Wayne State University, USA
Zixue Cheng	The University of Aizu, Japan
Ken Jen-Shiun Chiang	Tamkang University, Taiwan
Xavier Defago	JAIST, Japan
Lawrence Y. Deng	St. John's & Mary's Inst. of Tech., Taiwan
Mieso Denko,	University of Guelph, Canada
Marios D. Dikaiakos	University of Cyprus, Cyprus
Michael Ditze	University of Paderborn, Germany
Arjan Durrresi	Louisiana State University, USA
Frank Golatowski	University of Rostock, Germany
Takahiro Hara	Osaka University, Japan
Naohiro Hayashibara	Tokyo Denki University, Japan
Aiguo He	The University of Aizu, Japan
Pin-Han Ho	University of Waterloo, Canada
Hui-Huang Hsu	Tamkang University, Taiwan
Chung-Ming Huang	Nat. Cheng Kung Univ., Taiwan
Runhe Huang	Hosei University, Japan
Tsung-Chuan Huang	Nat. Sun Yat-sen Univ., Taiwan
Jason C. Hung	Northern Taiwan Inst. of Sci. and Tech., Taiwan
Ren-Hung Hwang	National Chung Cheng University, Taiwan
Jadwiga Indulska	Univ. of Queensland, Australia
Xiaohong Jiang	Tohoku University, Japan
Qun Jin	Waseda University, Japan
Chung-Ta King	National TsingHua University, Taiwan
Akio Koyama	Yamagata University, Japan
Stan Kurkovsky	Columbus State University, USA
Choonhwa Lee	Hanyang University, Korea
Wonjun Lee	Korea University, Korea
Hong-Va Leong	Hong Kong Polytechnic Univ., Hong Kong, China
Jiandong Li	Xidian University, China
Fuhua Oscar Lin	Athabasca University, Canada
Alex Zhaoyu Liu	Univ. of North Carolina at Charlotte, USA
Beniamino Di Martino	Second Univ. of Naples, Italy
Geyong Min	University of Bradford, UK
Yi Mu	University of Wollongong, Australia
Thomas Noel	University Louis Pasteur, France
Antonio Puliafito	University of Messina, Italy

**UISW 2005 Program/Technical Committee (continued)**

Aaron J. Quigley	University College Dublin, Ireland
Indrakshi Ray	Colorado State University, USA
Jae-cheol Ryou	Chungnam National Univ., Korea
Hiroki Saito	Tokyo Denki University, Japan
Kouichi Sakurai	Kyushu University, Japan
Elhadi Shakshuki	Acadia University, Canada
David Simplot-Ryl	Univ. Lille 1, France
Alexei Sourin	Nanyang Tech. Univ., Singapore
Ivan Stojmenovic	Ottawa University, Canada
Willy Susilo	University of Wollongong, Australia
Tsutomu Terada	Osaka University, Japan
Yu-Chee Tseng	National Chiao-Tung University, Taiwan
Javier Garcia Villalba	Complutense Univ. of Madrid, Spain
Cho-li Wang	Hong Kong University, Hong Kong, China
Li-Chun Wang	National Chiao-Tung University, Taiwan
Ying-Hong Wang	Tamkang University, Taiwan
Chaohui Wu	Zhejiang University, China
Jie Wu	Florida Atlantic University, USA
Bin Xiao	Hong Kong Polytechnic Univ., Hong Kong, China
Yang Xiao	University of Memphis, USA
Lu Yan	Turku Centre for Computer Science, Finland
Chu-Sing Yang	National Sun Yat-sen University, Taiwan
George Yee	National Research Council, Canada
Masao Yokota	Fukuoka Institute of Tech., Japan
Takaichi Yoshida	Kyushu Institute of Technology, Japan
Jon (Jong-Hoon) Youn	Univ. of Nebraska at Omaha, USA
Muhammed Younas	Oxford Brookes University, UK
Ming Yu	SUNY at Binghamton, USA
Salim Zabir	Panasonic R&D, Japan
Guozhen Zhang	Waseda University, Japan
Jingyuan (Alex) Zhang	University of Alabama, USA
Qiangfu Zhao	The University of Aizu, Japan
Xiaobo Zhou	University of Colorado at Colorado Springs, USA

## NCUS 2005 Executive Committee

General Chairs: Jingyuan (Alex) Zhang, University of Alabama, USA  
Jon (Jong-Hoon) Youn, University of Nebraska at Omaha,  
USA

Steering Chair: Laurence T. Yang, St. Francis Xavier University, Canada

Program Chairs: Lu Yan, Turku Centre for Computer Science (TUUS), Finland  
Luis Javier García Villalba, Complutense University of  
Madrid (UCM), Spain

## NCUS 2005 Program/Technical Committee

Nael Abu-Ghazaleh	SUNY at Binghamton, USA
Saad Biaz	Auburn University, USA
Jacir L. Bordim	University of Brasilia, Brazil
Phillip Bradford	University of Alabama, USA
Jiannong Cao	Hong Kong Polytechnic University, China
Guangbin Fan	University of Mississippi, USA
Satoshi Fujita	Hiroshima University, Japan
Xiaoyan Hong	University of Alabama, USA
Anup Kumar	University of Louisville, USA
Jiageng Li	University of West Georgia, USA
Koji Nakano	Hiroshima University, Japan
Huai-Rong Shao	Samsung, USA
Randy Smith	University of Alabama, USA
Dajin Wang	Montclair State University, USA
Zhijun Wang	University of Alabama, USA
Claudia Jacy Barenco Abbas	University of Brasilia, Brazil
Qing-An Zeng	University of Cincinnati, USA
Ming Yu	SUNY at Binghamton, USA
Jiang (Leo) Li	Howard University, USA
Mohamed Ould-Khaoua	University of Glasgow, UK
Mieso Denko	University of Guelph, Canada
Chih-Hung Chang	Tamkang University, Taiwan
Chung-Ta King	National TsingHua University, Taiwan
Yu-Chee Tseng	National Chiao-Tung University, Taiwan
Xinrong Zhou	Åbo Akademi, Finland
Antonio Puliafito	University of Messina, Italy

**NCUS 2005 Program/Technical Committee (continued)**

Chu-Sing Yang	National Sun Yat-sen University, Taiwan
Han-Chieh Chao	National Dong Hwa University, Taiwan
Jianhua Ma	Hosei University, Japan
Nidal Nasser	University of Guelph, Canada
Hong Shen	JAIST, Japan
Hai Jin	HUST, China
Doo-Hwan Bae	KAIST, Korea
Jun Pang	INRIA-Futurs, France
Rafael Timoteo de Sousa	Universidade de Brasilia, Brazil
Ricardo Puttini	Universidade de Brasilia, Brazil
Paulo Roberto de Lira Gondim	Universidade de Brasilia, Brazil
Mario Dantas	Universidade Federal de Santa Catarina, Brazil
Mirela S. M. A. Notare	Faculdades Barddal, Brazil
Alba Melo	Universidade de Brasilia, Brazil
Dan Grigoras	University College Cork, Ireland
Ami Marowka	Shenkar College of Engineering and Design, Israel
Hesham Ali	University of Nebraska at Omaha, USA
Chulho Won	University of Nebraska at Omaha, USA
Hamid Sharif	University of Nebraska at Lincoln, USA
Jitender Deogun	University of Nebraska at Lincoln, USA
Seungjin Park	Michigan Technological University, USA
Dana Petcu	Institute e-Austria Timisoara, Romania
Maria Ganzha	Private Institute of Higher Learning, Poland
Kathy Liszka	University of Akron, USA
Hyunjeong Lee	University of Nevada, USA
Song Ci	University of Massachusetts Boston, USA
Hyunyoung Lee	University of Denver, USA
Guanling Chen	University of Massachusetts Lowell, USA
Gary Marsden	University of Cape Town, South Africa
Marcin Paprzycki	Oklahoma State University, USA
Il Kyeun Ra	University of Colorado at Denver, USA

**SecUbiq 2005 Executive Committee**

General Chairs:	Edwin Sha, University of Texas at Dallas, USA Xiaobo Zhou, University of Colorado at Colorado Springs, USA
Program Chairs:	Alex Zhaoyu Liu, University of North Carolina at Charlotte, USA Bin Xiao, Hong Kong Polytechnic University, Hong Kong, China
Steering Chair:	Laurence T. Yang, St. Francis Xavier University, Canada

## SecUbiq 2005 Program/Technical Committee

Antonio Corradi	University of Bologna, Italy
Jemal Abawajy	Deakin University, Australia
Leemon Baird	US Air Force Academy, USA
John T. Brassil	HP Laboratories, USA
Yuanshun Dai	Indiana University-Purdue University, USA
Arjan Durrezi	Louisiana State University, USA
Hanping Hu	Huazhong University of Science and Technology, China
Hua Ji	Juniper Networks, USA
Zhiping Jia	Shangdong University, China
Zhen Jiang	West Chester University, USA
ShiGuang Ju	Jiangsu University, China
Seungjoo Kim	Sungkyunkwan University, Korea
Yoohwan Kim	University of Nevada, USA
Raymond Li	CISCO, USA
Sanglu Lu	Nanjing University, China
Jianhua Ma	Hosei University, Japan
Antonino Mazzeo	Second University of Naples, Italy
Jason A. Moore	US Air Force Academy, USA
Yi Mu	University of Wollongong, Australia
Mará S. Pérez-Hernández	Universidad Politécnica de Madrid, Spain
Xiao Qin	New Mexico Institute of Mining and Technology, USA
Jae-cheol Ryou	Chungnam National University, Korea
Kouichi Sakurai	Kyushu University, Japan
Dino Schweitzer	US Air Force Academy, USA
Chi-Sheng(Daniel) Shih	National Taiwan University, Taiwan
Xinmei Wang	Xidian University, China
Chuan-Kun Wu	Chinese Academy of Sciences, China
Yang Xiao	The University of Memphis, USA
Ming Xu	National University of Defence Technology, China
George Yee	National Research Council, Canada
Meng Yu	Monmouth University, USA
Ning Zhang	University of Manchester, UK
Xukai Zou	Indiana University-Purdue University, USA

## SecUbiq 2005 Reviewers

Jemal Abawajy	Yuanshun Dai	Hua Ji
Leemon Baird	Arjan Durrezi	Zhiping Jia
John T. Brassil	Hanping Hu	Zhen Jiang

ShiGuang Ju	Yi Mu	Chuan-Kun Wu
Seungjoo Kim	Mará S. Pérez–Hernández	Yang Xiao
YooHwan Kim	Xiao Qin	Ming Xu
Raymond Li	Jae-cheol Ryou	George Yee
Sanglu Lu	Kouchi Sakurai	Meng Yu
Jianhua Ma	Dino Schweitzer	Ning Zhang
Antonino Mazzeo	Chi-Sheng(Daniel) Shih	Xukai Zou
Jason A. Moore	Xinmei Wang	

## USN 2005 Executive Committee

Program Chair:	Daeyoung Kim, Information and Communications University, Korea
Steering Committee:	Jongsuk Chae, ETRI, Korea Joongsoo Ma, Information and Communications University, Korea Hao Min, Fudan University, China Kang Shin, University of Michigan, USA Yu-Chee Tseng, National Chiao Tung Univ., Taiwan

## USN 2005 Program/Technical Committee

Yunju Baek	Pusan National University, Korea
Chih-Yung Chang	Tamkang University, Taiwan
Paul Chartier	Praxis Consultants, UK
Yuh-Shyan Chen	National Chung Cheng University, Taiwan
Yoonmee Doh	ETRI, Korea
Anthony Furness	AIMUK/UCE, UK
Paul Havinga	University of Twente, Netherlands
Yan Huang	Motorola Labs, USA
Rajgopal Kannan	LSU, USA
Chung-Ta King	National Tsing-Hua University, Taiwan
Youngbae Ko	Ajou University, Korea
Noboru Koshizuka	University of Tokyo, Japan
Bhaskar Krishnamachari	USC, USA
Koen Langendoen	Delft University of Technology, Netherlands
Insup Lee	University of Pennsylvania, USA
Sungyoung Lee	Kyunghee University, Korea
Yann-Hang Lee	Arizona State University, USA
Wei Lou	Hong Kong Polytechnic University, China
Wen-Chih Peng	National Chiao Tung Univ., Taiwan
Sang Son	University of Virginia, USA
Mohamed Younis	University of Maryland, USA
Chansu Yu	Cleveland State University, USA
Feng Zhao	Microsoft, USA

## **TAUES 2005 Executive Committee**

- General Chairs: Yuan-Shun Dai, Indiana University-Purdue  
University, USA  
Manish Parashar, Rutgers University, USA
- Program Chairs: Roy Sterritt, University of Ulster, N. Ireland  
Xukai Zou, Indiana University-Purdue University, USA  
Xiao Qin, New Mexico Inst. of Mining and  
Technology, USA
- Steering Chairs: Laurence T. Yang, St. Francis Xavier University,  
Canada  
Jianhua Ma, Hosei University, Japan
- Advisory Chair: Salim Hariri, University of Arizona Tucson, USA



**TAUES 2005 Program/Steering/Advisory Committees**

Jingde Cheng	Saitama University, Japan
Makoto Takizawa	Tokyo Denki University, Japan
Jogesh Muppala	Hong Kong University of Science and Technology, China
Mike Hinchey	NASA Goddard Flight Center, MD, USA
Tadashi Dohi	Hiroshima University, Japan
Leonard Barolli	Fukuoka Institute of Technology, Japan
Gregory Levitin	Technion-Israel Institute of Technology, Israel
Haruhisa Ichikawa	NTT Network Innovation Lab, Japan
Dave Bustard	University of Ulster, UK
Beniamino Di Martino	Second University of Naples, Italy
Umberto Villano	University of Sannio, Italy
Jemal Abawajy	Deakin University, Australia
Phillip Bradford	University of Alabama, USA
Petre Dini	Cisco Systems Inc., USA
Michael Ditze	University of Paderborn, Germany
David P. Duggan	Sandia National Laboratories, Sandia, USA
Katerina Goseva-Popstojanova	West Virginia University, USA
Chandana Gamage	Free University, Netherlands
Sachin Garg	AVAYA Labs, USA
Frank Golatowski	University of Rostock, Germany
Swapna Gokhale	University of Connecticut, USA
Michael Grottke	Duke University, USA
Minaxi Gupta	Indiana University, USA
Gail Kaiser	Columbia University, USA
Andrew Laws	Liverpool John Moores University, UK
Xiaolin Li	Rutgers University, USA
Hua Liu	Rutgers University, USA
Man Lin	St. Francis Xavier University, Canada
Rodrigo de Mello	University of Sao Paulo, Brazil
Maria S. Perez-Hernandez	Universidad Politécnica de Madrid, Spain
Rami Melhem	University of Pittsburgh, USA
Adel Rouz	Fujitsu, UK
Biplab K. Sarker	University of New Brunswick, Canada
Elhadi Shakshuki	Acadia University, Canada
Michael Smirnov	Fraunhofer Institute FOKUS, German
Kalyan Vaidyanathan	Sun Microsystems, USA
Bin Xiao	Hong Kong Polytechnic University, China
Jiang (Linda) Xie	University of North Carolina at Charlotte, USA
Liudong Xing	University of Massachusetts Dartmouth, USA
Xun Yi	Victoria University, Australia
Tomoya Enokido	Rissho University, Japan

## XVIII Organization

Hao Yin

Bo Yang

Ming Zhu

Tsinghua University, China

University of Electronic Sci. and  
Tech. of China, China

Oracle, USA

## TAUES 2005 Reviewers

Dave Bustard

Beniamino Di Martino

Umberto Villano

Jemal Abawajy

Phillip Bradford

Petre Dini

Michael Ditze

David P. Duggan

K. Goseva-Popstojanova

Chandana Gamage

Sachin Garg

Frank Golatowski

Swapna Gokhale

Michael Grottke

Minaxi Gupta

Gail Kaiser

Andrew Laws

Xiaolin Li

Hua Liu

Man Lin

Rodrigo de Mello

M.S. Perez-Hernandez

Rami Melhem

Biplab K. Sarker

Elhadi Shakshuki

Michael Smirnov

Kalyan Vaidyanathan

Bin Xiao

Jiang (Linda) Xie

Liudong Xing

Xun Yi

Tomoya Enokido

Hao Yin

Bo Yang

Ming Zhu

# Table of Contents

## The Second International Symposium on Ubiquitous Intelligence and Smart Worlds (UISW 2005)

### Session 1: Smart Environments and Systems I

Human Activity Recognition Based on Surrounding Things <i>Naoharu Yamada, Kenji Sakamoto, Goro Kunito, Kenichi Yamazaki, Satoshi Tanaka</i> .....	1
Baton: A Service Management System for Coordinating Smart Things in Smart Spaces <i>Jingyu Li, Yuanchun Shi</i> .....	11
An Extensible Ubiquitous Architecture for Networked Devices in Smart Living Environments <i>Thierry Bodhuin, Gerardo Canfora, Rosa Preziosi, Maria Tortorella</i> .....	21
A World Model for Smart Spaces <i>Ichiro Satoh</i> .....	31

### Session 2: Agent Based Smart Computing

Dealing with Emotional Factors in Agent Based Ubiquitous Group Decision <i>Goreti Marreiros, Carlos Ramos, José Neves</i> .....	41
A Multi-agent Software Platform Accommodating Location-Awareness for Smart Space <i>Hongliang Gu, Yuanchun Shi, Guangyou Xu, Yu Chen</i> .....	51
Context-Aware Ubiquitous Data Mining Based Agent Model for Intersection Safety <i>Flora Dilys Salim, Shonali Krishnaswamy, Seng Wai Loke, Andry Rakotonirainy</i> .....	61
Development of Knowledge-Filtering Agent Along with User Context in Ubiquitous Environment <i>Takao Takenouchi, Takahiro Kawamura, Akihiko Ohsuga</i> .....	71

### Session 3: Smart Computing Environments

Application-Driven Customization of an Embedded Java Virtual Machine <i>Alexandre Courbot, Gilles Grimaud, Jean-Jacques Vandewalle, David Simplot-Ryl</i> .....	81
A Study on Fast JCVM with New Transaction Mechanism and Caching-Buffer Based on Java Card Objects with a High Locality <i>Min-Sik Jin, Won-Ho Choi, Yoon-Sim Yang, Min-Soo Jung</i> .....	91
Intelligent Object Extraction Algorithm Based on Foreground/Background Classification <i>Jhing-Fa Wang, Han-Jen Hsu, Jyun-Sian Li</i> .....	101
Thin Client Based User Terminal Architecture for Ubiquitous Computing Environment <i>Tatsuo Takahashi, Satoshi Tanaka, Kenichi Yamazaki, Tadanori Mizuno</i> .....	111
An Application Development Environment for Rule-Based I/O Control Devices <i>Ryohei Sagara, Yasue Kishino, Tsutomu Terada, Tomoki Yoshihisa, Masahiko Tsukamoto, Shojiro Nishio</i> .....	121
A uWDL Handler for Context-Aware Workflow Services in Ubiquitous Computing Environments <i>Yongyun Cho, Joohyun Han, Jaeyoung Choi, Chae-Woo Yoo</i> .....	131

### Session 4: Smart Environments and Systems II

SMMART, a Context-Aware Mobile Marketing Application: Experiences and Lessons <i>Stan Kurkovsky, Vladimir Zanev, Anatoly Kurkovsky</i> .....	141
Ubiquitous Organizational Information Service Framework for Large Scale Intelligent Environments <i>Kwang-il Hwang, Won-hee Lee, Seok-hwan Kim, Doo-seop Eom, Kyeong Hur</i> .....	151
TS-U: Temporal-Spatial Methodology for Application Checking of the Systems in the Ubiquitous Environment <i>Fran Jarnjak, Jinhyung Kim, Yixin Jing, Hoh Peter In, Dongwon Jeong, Doo-Kwon Baik</i> .....	161

Ubiquitous Learning on Pocket SCORM <i>Hsuan-Pu Chang, Wen-Chih Chang, Yun-Long Sie, Nigel H. Lin, Chun-Hong Huang, Timothy K. Shih, Qun Jin</i> .....	171
An Application Based on Spatial-Relationship to Basketball Defensive Strategies <i>Su-Li Chin, Chun-Hong Huang, Chia-Tong Tang, Jason C. Hung</i> .....	180
Intrinsically Motivated Intelligent Rooms <i>Owen Macindoe, Mary Lou Maher</i> .....	189
<b>Session 5: Smart Networking and Protocols I</b>	
Multivariate Stream Data Reduction in Sensor Network Applications <i>Sungbo Seo, Jaewoo Kang, Keun Ho Ryu</i> .....	198
Implementing a Graph Neuron Array for Pattern Recognition Within Unstructured Wireless Sensor Networks <i>M. Baqer, A.I. Khan, Z.A. Baig</i> .....	208
Building Graphical Model Based System in Sensor Networks <i>Dongyu Shi, Jinyuan You, Zhengwei Qi</i> .....	218
Energy-Aware Broadcasting Method for Wireless Sensor Network <i>Cheol-Min Park, Dae-Won Kim, Jun Hwang</i> .....	228
Anonymous Routing in Wireless Mobile Ad Hoc Networks to Prevent Location Disclosure Attacks <i>Arjan Duresi, Vamsi Paruchuri, Mimoza Duresi, Leonard Barolli</i> .....	238
<b>Session 6: Smart Environments and Systems III</b>	
The Design and Implementation of a Location-Aware Service Bundle Manager in Smart Space Environments <i>Minwoo Son, Soonyong Choi, Dongil Shin, Dongkyoo Shin</i> .....	248
A Context-Aware and Augmented Reality-Supported Service Framework in Ubiquitous Environments <i>Jae Yeol Lee, Dong Woo Seo</i> .....	258

A Smart Method of Cooperative Learning Including Distant Lectures and Its Experimental Evaluations <i>Dilmurat Tilwaldi, Toshiya Takahashi, Yuichiro Mishima, Jun Sawamoto, Hisao Koizumi</i> .....	268
u-KoMIPS: A Medical Image Processing System in a Ubiquitous Environment <i>Soo Jin Lee, Moon Hae Kim</i> .....	278
The Extended PARLAY X for an Adaptive Context-Aware Personalized Service in a Ubiquitous Computing Environment <i>Sungjune Hong, Sunyoung Han, Kwanho Song</i> .....	288
A Context-Aware System for Smart Home Applications <i>Wen-Yang Wang, Chih-Chieh Chuang, Yu-Sheng Lai, Ying-Hong Wang</i> .....	298
 <b>Session 7: Smart Computing</b>	
Human Position/Height Detection Using Analog Type Pyroelectric Sensors <i>Shinya Okuda, Shigeo Kaneda, Hirohide Haga</i> .....	306
ENME: An ENriched MEDIA Application Utilizing Context for Session Mobility; Technical and Human Issues <i>Egil C. Østhus, Per-Oddvar Osland, Lill Kristiansen</i> .....	316
DartDataFlow: Semantic-Based Sensor Grid <i>Huajun Chen, Zhiyong Ye, Zhaohui Wu</i> .....	326
Sentient Artefacts: Acquiring User's Context Through Daily Objects <i>Kaori Fujinami, Tatsuo Nakajima</i> .....	335
A Multi-dimensional Model for Task Representation and Allocation in Intelligent Environments <i>Victor Zamudio, Vic Callaghan, Jeannette Chin</i> .....	345
Norms Enforcement as a Coordination Strategy in Ubiquitous Environments <i>Ismail Khalil Ibrahim, Reinhard Kronsteiner, Gabriele Kotsis</i> .....	355

## Session 8: Smart Objects

A Java-Based RFID Service Framework with Semantic Data Binding Between Real and Cyber Spaces <i>Kei Nakanishi, Makoto Setozaki, Jianhua Ma, Runhe Huang</i> . . . . .	365
Kallima: A Tag-Reader Protocol for Privacy Enhanced RFID System <i>Yusuke Doi, Shirou Wakayama, Masahiro Ishiyama, Satoshi Ozaki, Atsushi Inoue</i> . . . . .	375
Selective Collision Based Medium Access Control Protocol for Proactive Protection of Privacy for RFID <i>JuSung Park, Jeonil Kang, DaeHun Nyang</i> . . . . .	383
<i>iCane</i> – A Partner for the Visually Impaired <i>Tsung-Hsiang Chang, Chien-Ju Ho, David Chawei Hsu, Yuan-Hsiang Lee, Min-Shieh Tsai, Mu-Chun Wang, Jane Hsu</i> . . . . .	393

## Session 9: Security and Fault Tolerance of Smart Systems I

ORAIID: An Intelligent and Fault-Tolerant Object Storage Device <i>Dan Feng, Lingfang Zeng, Fang Wang, Shunda Zhang</i> . . . . .	403
Architecture Based Approach to Adaptable Fault Tolerance in Distributed Object-Oriented Computing <i>Rodrigo Lanka, Kentaro Oda, Takaichi Yoshida</i> . . . . .	413
Security Analysis of Michael: The IEEE 802.11i Message Integrity Code <i>Jianyong Huang, Jennifer Seberry, Willy Susilo, Martin Bunder</i> . . . . .	423
A Framework for Protecting Private Information Through User-Trusted-Program and Its Realizability <i>Ken'ichi Takahashi, Kouichi Sakurai, Makoto Amamiya</i> . . . . .	433

## Session 10: Smart Networking and Protocols II

Performance Analysis of IP Micro-mobility Protocols in Single and Simultaneous Movements Scenario <i>Giuseppe De Marco, S. Loreto, Leonard Barolli</i> . . . . .	443
--	-----

HMRP: Hierarchy-Based Multipath Routing Protocol for Wireless Sensor Networks  
*Ying-Hong Wang, Hung-Jen Mao, Chih-Hsiao Tsai, Chih-Chieh Chuang* ..... 452

On Energy-Aware Dynamic Clustering for Hierarchical Sensor Networks  
*Joongheon Kim, Wonjun Lee, Eunkyo Kim, Joonmo Kim, Choonhwa Lee, Sungjin Kim, Sooyeon Kim*..... 460

Neighbor Node Discovery Algorithm for Energy-Efficient Clustering in Ubiquitous Sensor Networks  
*Ji Young Choi, Chung Gu Kang, Yong Suk Kim, Kyeong Hur* ..... 470

**Session 11: Security and Fault Tolerance of Smart Systems II**

A Short Proxy Signature Scheme: Efficient Authentication in the Ubiquitous World  
*Xinyi Huang, Yi Mu, Willy Susilo, Fangguo Zhang, Xiaofeng Chen* ..... 480

The Design and Implementation of Secure Event Manager Using SPKI/SDSI Certificate  
*YoungLok Lee, HyungHyo Lee, Seungyong Lee, HeeMan Park, BongNam Noh* ..... 490

Object Reminder and Safety Alarm  
*Chi-yau Lin, Chia-nan Ke, Shao-you Cheng, Jane Yung-jen Hsu, Hao-hua Chu* ..... 499

Synchronization and Recovery in an Embedded Database System for Read-Only Transactions  
*Subhash Bhalla, Masaki Hasegawa* ..... 509

**Session 12: Intelligent Computing**

Learning with Data Streams – An NNTree Based Approach  
*Qiangfu Zhao* ..... 519

Generating Smart Robot Controllers Through Co-evolution  
*Kouichi Sakamoto, Qiangfu Zhao* ..... 529



Integrated Multimedia Understanding for Ubiquitous Intelligence Based on Mental Image Directed Semantic Theory <i>Masao Yokota, Genci Capi</i> .....	538
---	-----

### Session 13: Smart Environments and Systems IV

Hyper-Interactive Video Browsing by a Remote Controller and Hand Gestures <i>Hui-Huang Hsu, Timothy K. Shih, Han-Bin Chang, Yi-Chun Liao, Chia-Tong Tang</i> .....	547
---	-----

Mobile Computing with MPEG-21 <i>Marios C. Angelides, Anastasis A. Sofokleous, Christos N. Schizas</i> ..	556
--	-----

A Unified Context Model: Bringing Probabilistic Models to Context Ontology <i>Binh An Truong, YoungKoo Lee, Sung Young Lee</i> .....	566
---	-----

### IFIP International Symposium on Network-Centric Ubiquitous Systems (NCUS 2005)

A Component-Based Adaptive Model for Context-Awareness in Ubiquitous Computing <i>Soo-Joong Ghim, Yong-Ik Yoon, Ilkyeun Ra</i> .....	576
---	-----

Improvement of an Efficient User Identification Scheme Based on ID-Based Cryptosystem <i>Eun-Jun Yoon, Kee-Young Yoo</i> .....	586
---	-----

The Wrong Challenge of 'Pervasive Computing': The Paradigm of Sensor-Based Context-Awareness <i>Eric Angaman</i> .....	592
---	-----

An Abstract Model for Incentive-Enhanced Trust in P2P Networks <i>Mats Neovius</i> .....	602
---	-----

Construction of Credible Ubiquitous P2P Content Exchange Communities <i>Yuki Yokohata, Hiroshi Sunaga, Hiroyuki Nakamura</i> .....	612
---	-----

Location-Based Routing Protocol for Energy Efficiency in Wireless Sensor Networks <i>Hyuntae Cho, Yunju Baek</i> .....	622
---	-----

Efficient Access of Remote Resources in Embedded Networked Computer Systems <i>Paul S. Usher, Neil C. Audsley</i> .....	632
Lightweight Ontology-Driven Representations in Pervasive Computing <i>Jarostaw Domaszewicz, Michal Rój</i> .....	642
Object Tracking Using Durative Events <i>Eiko Yoneki, Jean Bacon</i> .....	652
Design of Integrated Routing System for Mobile Ad Hoc Networks Overlaying Peer-to-Peer Applications <i>Yan Annie Ding, David Everitt</i> .....	663
A Design of Privacy Conscious RFID System Using Customizing Privacy Policy Based Access Control <i>Byungil Lee, Howon Kim</i> .....	673
Efficient Resource Management Scheme of TCP Buffer Tuned Parallel Stream to Optimize System Performance <i>Kun Myon Choi, Ewi-Nam Huh, Hyunseung Choo</i> .....	683
Multi-level Service Differentiation Scheme for the IEEE 802.15.4 Sensor Networks <i>Euijik Kim, Meejoung Kim, Sungkwan Youm, Seokhoon Choi, Chul-Hee Kang</i> .....	693
Development of Event Manager and Its Application in Jini Environment <i>YoungLok Lee, HyungHyo Lee, Seungyong Lee, InSu Kim, BongNam Noh</i> .....	704
On Scalability and Mobility Management of Hierarchical Large-Scale Ad Hoc Networks <i>Ming-Hui Tsai, Tzu-Chiang Chiang, Yueh-Min Huang</i> .....	714
Exploring Small-World-Like Topologies Via SplitProber: Turning Power Laws into an Advantage in Unstructured Overlays <i>Xinli Huang, Wenju Zhang, Fanyuan Ma, Yin Li</i> .....	724
Efficient Uplink Scheduler Architecture of Subscriber Station in IEEE 802.16 System <i>Woo-Jae Kim, Joo-Young Baek, Sun-Don Lee, Young-Joo Suh, Yun-Sung Kim, Jin-A Kim</i> .....	734
A Survey of Anonymous Peer-to-Peer File-Sharing <i>Tom Chothia, Konstantinos Chatzizokolakis</i> .....	744

A Churn-Resistant Strategy for a Highly Reliable P2P System <i>Giscard Wepiwé, Sahin Albayrak</i> .....	756
Building a Peer-to-Peer Overlay for Efficient Routing and Low Maintenance <i>Honghao Wang, Yiming Hu</i> .....	766
Dynamic Object Assignment in Object-Based Storage Devices <i>Lingjun Qin, Dan Feng</i> .....	776
Dynamic Resource Discovery for Sensor Networks <i>Sameer Tilak, Kenneth Chiu, Nael B. Abu-Ghazaleh, Tony Fountain</i> .....	785
Survey on Location Authentication Protocols and Spatial-Temporal Attestation Services <i>A.I. González-Tablas, K. Kursawe, B. Ramos, A. Ribagorda</i> .....	797
Predicate Detection Using Event Streams in Ubiquitous Environments <i>Ajay D. Kshemkalyani</i> .....	807
<b>The First International Workshop on Security in Ubiquitous Computing Systems (SecUbiq 2005)</b>	
Image Watermarking Technique Based on Two-Dimensional Chaotic Stream Encryption <i>Hanping Hu, Yongqiang Chen</i> .....	817
Identity-Based Universal Designated Verifier Signatures <i>Fanguo Zhang, Willy Susilo, Yi Mu, Xiaofeng Chen</i> .....	825
Short Designated Verifier Proxy Signature from Pairings <i>Xinyi Huang, Yi Mu, Willy Susilo, Futai Zhang</i> .....	835
An Embedded Gateway Based on Real-Time Database <i>Zhiping Jia, Xinxiao Qiao</i> .....	845
Efficient Authentication Scheme for Routing in Mobile Ad Hoc Networks <i>Shidi Xu, Yi Mu, Willy Susilo</i> .....	854
Collision Attack on XTR and a Countermeasure with a Fixed Pattern <i>Dong-Guk Han, Tsuyoshi Takagi, Tae Hyun Kim, Ho Won Kim, Kyo Il Chung</i> .....	864

Security in Persistently Reactive Systems <i>Takumi Endo, Junichi Miura, Koichi Nanashima, Shoichi Morimoto, Yuichi Goto, Jingde Cheng</i> . . . . .	874
ID-Based Access Control and Authority Delegations <i>So-Young Park, Sang-Ho Lee</i> . . . . .	884
How to Construct Secure Cryptographic Location-Based Services <i>Jun Anzai, Tsutomu Matsumoto</i> . . . . .	894
A Key Management Scheme for Mobile Ad Hoc Networks Based on Threshold Cryptography for Providing Fast Authentication and Low Signaling Load <i>Hoang Nam Nguyen, Hiroaki Morino</i> . . . . .	905
Program Obfuscation Scheme Using Random Numbers to Complicate Control Flow <i>Tatsuya Toyofuku, Toshihiro Tabata, Kouichi Sakurai</i> . . . . .	916
Authenticated Public Key Distribution Scheme Without Trusted Third Party <i>Jae Hyung Koo, Bum Han Kim, Dong Hoon Lee</i> . . . . .	926
Cryptanalysis of a Generalized Anonymous Buyer-Seller Watermarking Protocol of IWDW 2004 <i>Bok-Min Goi, Raphael C.-W. Phan, M.U. Siddiqi</i> . . . . .	936
Efficient RFID Authentication Protocol for Ubiquitous Computing Environment <i>Eun Young Choi, Su Mi Lee, Dong Hoon Lee</i> . . . . .	945
A New Simple Authenticated Key Agreement and Protected Password Change Protocol <i>Eun-Jun Yoon, Kee-Young Yoo</i> . . . . .	955
A Method for Deciding Quantization Steps in QIM Watermarking Schemes <i>Yunho Lee, Kwangwoo Lee, Seungjoo Kim, Dongho Won, Hyungkyu Yang</i> . . . . .	965
A New DDoS Detection Model Using Multiple SVMs and TRA <i>Jungtaek Seo, Cheolho Lee, Taeshik Shon, Kyu-Hyung Cho, Jongsub Moon</i> . . . . .	976

PPF Model with CTNT to Defend Web Server from DDoS Attack <i>Jungtaek Seo, Cheolho Lee, Jungtae Kim, Taeshik Shon, Jongsub Moon</i> .....	986
Efficient Key Agreement for Merging Clusters in Ad-Hoc Networking Environments <i>Sooyeon Shin, Taekyoung Kwon</i> .....	996
An Effective Method for Location Privacy in Ubiquitous Computing <i>Gunhee Lee, Wonil Kim, Dong-kyoo Kim</i> .....	1006
Integrated Support for Location Aware Security Services in Enterprise Wireless Networks <i>Zhaoyu Liu, Peeyush Sharma, Jian Raymond Li</i> .....	1016

## **The 1st International Workshop on RFID and Ubiquitous Sensor Networks (USN 2005)**

### **Session 1: RFID**

Optimal Scheduling for Networks of RFID Readers <i>Vinay Deolalikar, John Recker, Malena Mesarina, Salil Pradhan</i> .....	1025
PULSE: A MAC Protocol for RFID Networks <i>Shailesh M. Birari, Sridhar Iyer</i> .....	1036
<i>RFIDcover</i> - A Coverage Planning Tool for RFID Networks with Mobile Readers <i>S. Anusha, Sridhar Iyer</i> .....	1047
Vibration Powered Battery-Assisted Passive RFID Tag <i>Elaine Lai, Andrew Redfern, Paul Wright</i> .....	1058
Wireless RFID Networks for Real-Time Customer Relationship Management <i>Philipp Schloter, Hamid Aghajan</i> .....	1069
Tree-Based Classification Algorithm for Heterogeneous Unique Item ID Schemes <i>Yong Hwan Lee, Hee Jung Kim, Byeong-hee Roh, S.W. Yoo, Y.C. Oh</i> .....	1078

An Efficient Key Pre-distribution Scheme for Secure Distributed Sensor Networks  
*Sung Jin Choi, Hee Yong Youn* . . . . . 1088

**Session 2: USN**

Energy-Driven Adaptive Clustering Hierarchy (EDACH) for Wireless Sensor Networks  
*Kyung Tae Kim, Hee Yong Youn* . . . . . 1098

A Load-Balancing and Energy-Aware Clustering Algorithm in Wireless Ad-Hoc Networks  
*Wang Jin, Shu Lei, Jinsung Cho, Young-Koo Lee, Sungyoung Lee, Yonil Zhong* . . . . . 1108

Energy-Efficient Cluster Reconfiguration with Fair Cluster Formations in Sensor Networks  
*Hyang-tack Lee, Yong-hyun Jo, Byeong-hee Roh, S.W. Yoo* . . . . . 1118

Virtual Sink Rotation: Low-Energy Scalable Routing Protocol for Ubiquitous Sensor Networks  
*Lynn Choi, Kwangseok Choi, Jungsun Kim, Byung Joon Park* . . . . . 1128

FERMA: An Efficient Geocasting Protocol for Wireless Sensor Networks with Multiple Target Regions  
*Young-Mi Song, Sung-Hee Lee, Young-Bae Ko* . . . . . 1138

Power-Aware Position Vector Routing for Wireless Sensor Networks  
*Sangsoo Lee, Daeyoung Kim, Sungjin Ahn, Noseong Park* . . . . . 1148

Multicast Routing with Minimum Energy Cost and Minimum Delay in Wireless Sensor Networks  
*Zhao Li, Wei Zhang, Hengchang Liu, Baohua Zhao, Yugui Qu* . . . . . 1157

Self Location Estimation Scheme Using ROA in Wireless Sensor Networks  
*Yun Kyung Lee, Eui Hyeok Kwon, Jae Sung Lim* . . . . . 1169

Energy-Efficient Target Localization Based on a Prediction Model  
*Yu Gu, Wei Zhang, HengChang Liu, Baohua Zhao, Yugui Qu* . . . . . 1178

Reducing Congestion in Real-Time Multi-party-tracking Sensor Network Applications  
*Wonwoo Jung, Sujeong Shin, Sukwon Choi, Hojung Cha* . . . . . 1191

Variable-Radii Method Based on Probing Mechanism (VRPM): An Energy Conservation Method for Wireless Active Sensor Networks <i>Qi Zhou, Takuya Asaka, Tatsuro Takahashi</i> .....	1201
---	------

## **The International Workshop on Trusted and Autonomic Ubiquitous and Embedded Systems (TAUES 2005)**

### **Session 1: Autonomic Computing**

Automata-Theoretic Performance Analysis Method of Soft Real-Time Systems <i>Satoshi Yamane</i> .....	1211
A Component-based Architecture for an Autonomic Middleware Enabling Mobile Access to Grid Infrastructure <i>Ali Sajjad, Hassan Jameel, Umar Kalim, Young-Koo Lee, Sungyoung Lee</i> .....	1225
Autonomic Agents for Survivable Security Systems <i>Roy Sterritt, Grainne Garrity, Edward Hanna, Patricia O'Hagan</i> ...	1235
Towards Formal Specification and Generation of Autonomic Policies <i>Roy Sterritt, Michael G. Hinchey, James L. Rash, Walt Truszkowski, Christopher A. Rouff, Denis Gracanin</i> .....	1245

### **Session 2: Security**

Intrusion Detection with CUSUM for TCP-based DDoS <i>Fang-Yie Leu, Wei-Jie Yang</i> .....	1255
A Digital Content Distribution Using a Group-key and Multi-layered Structure Based on Web <i>Yun-Ji Na, Il Seok Ko</i> .....	1265
Access Policy Sheet for Access Control in Fine-Grained XML <i>Jing Wu, Yi Mu, Jennifer Seberry, Chun Ruan</i> .....	1273

### **Session 3: Dependable Computing**

Monitoring the Health Condition of a Ubiquitous System: Rejuvenation vs. Recovery <i>Kazuki Iwamoto, Tadashi Dohi, Naoto Kaio</i> .....	1283
--	------

A Dependability Management Mechanism for Ubiquitous Computing Systems <i>Changyeol Choi, Sungsoo Kim</i> . . . . .	1293
Reassignment Scheme of an RFID Tag's Key for Owner Transfer <i>Junichiro Saito, Kenji Imamoto, Kouichi Sakurai</i> . . . . .	1303
<b>Author Index</b> . . . . .	1313



# Human Activity Recognition Based on Surrounding Things

Naoharu Yamada, Kenji Sakamoto, Goro Kunito,  
Kenichi Yamazaki, and Satoshi Tanaka

Network Laboratories, NTT DoCoMo, Inc.,  
3-5 Hikarino-oka, Yokosuka city, Kanagawa 239-8536, Japan  
{yamada, sakamoto, kunito, yamazaki,  
satoshi}@netlab.nttdocomo.co.jp

**Abstract.** This paper proposes human activity recognition based on the actual semantics of the human's current location. Since predefining the semantics of location is inadequate to identify human activities, we process information about things to automatically identify the semantics based on the concept of affordance. Ontology is used to deal with the various possible representations of things detected by RFIDs, and a multi-class Naïve Bayesian approach is used to detect multiple actual semantics from the terms representing things. Our approach is suitable for automatically detecting possible activities under a variety of characteristics of things including polysemy and variability. Preliminary experiments on manually collected datasets of things demonstrated its noise tolerance and ability to rapidly detect multiple actual semantics from existing things.

## 1 Introduction

Owing to the downsizing and increasing sophistication of computing appliances, the Ubiquitous Computing Environment proposed by Mark Weiser [22] is becoming reality. In the Ubiquitous Computing Environment, people will enjoy new services called "ubiquitous services". The appropriate ubiquitous services are provided depending on user's activities. While traditional services are reactive and uniform for every user, ubiquitous services are proactive and adaptive to each user. For example, when a user is shopping in a food court, the system can tell him what is in his refrigerator and what is missing. When the user unintentionally leaves his umbrella in a shop or train, the system detects the omission and informs the user. One of the essential issues in achieving ubiquitous services is how to recognize human activities since services are provided depending on the user's activities not his explicit requests. If system misjudges the activity, the user will receive useless and annoying services.

This paper presents a method to infer human activities based on the actual semantics of the human's current location. We name it activity space. Activity spaces (AS) are the logically defined spaces in which the user will perform a particular activity. By identifying activity space at user's current position, the system can infer the user's activities. Activity space is characterized by continual creation and disappearance. For example, when a flea market is held at a park, a *shopping AS* only exists during the period of the flea market. Therefore, we need a way of detecting the existence of

activity spaces automatically. To solve this issue, we focus on “things” that compose activity spaces. Since things basically have the purpose of existence and affordance [4] that offers people to do with them, they can specify human activities. Each thing can be identified by a Radio Frequency Identification (RFID) tag. Since various terms can be used to represent the same thing, we acquire all representations by utilizing ontology [5]. Activity spaces are detected by employing a topic detection method designed for document handling because we can draw an analogy between documents composed by words and activity spaces composed by things. Preliminary experiments utilizing actual things data demonstrate the feasibility of our proposed method.

## 2 Related Work

Approaches to tackle the essential issue of human activity recognition are classified into two types. One is utilizing wearable sensors [14] and the other is utilizing various sensors attached to things in the environment. Though the former approach is appropriate for fundamental activities such as sitting, standing, and walking by detecting limb motions, it places a burden on users since they wear the devices. For the latter approach, some papers focused on detecting the interaction between humans and objects by utilizing a camera [7] or an ultrasonic sensor [11]. Unfortunately, implementation costs are very high and the approaches only work in the laboratory. Tapia et al. [17] developed environmental state change sensors and Fishkin et al. [3] utilized RFID tags. They recognize user’s activities based on sequential data of things that the user touches or grasps. They achieved low-cost implementation and their work are applicable to real world environments. However, it is difficult for them to identify the activities including many non-sequential interactions with various things since the user may perform these activities in various ways.

Our main idea is to focus on activity spaces in identifying human activities. The simple approach is inference based on the predefined semantics of location such as a map [13] [21]. They focused more on how to identify the spatial position of users and less on how to specify the semantics of the spatial position because they assumed that the semantics of a spatial position was static. However, the effectiveness of this approach is limited since the activities that a location can offer are fixed and some locations do not specify just one activity. As for the former, the location semantics can change over time such as a flea market. However, these semantics are not handled at all in prior works. Typical examples of the latter locations are *a living room* at home or a *multipurpose room*. Though *a living room* can support various activities such as *studying, working, eating, and playing TV games*, the system cannot identify the activities actually supported by a room since it depends on the equipment in the room.

Our approach is to focus on the things forming the user’s immediate environment to identify activity spaces. Moreover, things existing in a certain space can be easily detected by RFID tags. They are seen as replacing the barcode in the area of logistics. Some companies or governments now require suppliers to attach RFID tags to every item. EPC Global [2] and Ubiquitous ID center [19] have proposed an ID scheme that makes it possible to put a unique serial number on every item. Considering this background, we can assume that everything will soon have its own RFID tag. This means that RFID tags are the most promising approach to realizing the Ubiquitous Computing Environment. The use of RFID tags demands the use of RFID tag readers.

They are located in the environment or a user carries one. From the perspective of hardware cost, there is tradeoff between these two methods: if the target space is large and the number of users is small, the former method is better, and otherwise, the latter is better. In this paper, we assume RFID tag readers are placed in the environment.

### 3 Activity Space Detection

This section clarifies the definition, characteristics, and technical issues of an activity space. It then describes the proposed approach based on ontology and multi-class Naïve Bayes for automatic detection of activity spaces.

#### 3.1 Activity Space: Definition, Characteristics, and Technical Issues

An activity space is a logically defined space that affords the user some particular activity. Examples are “shopping AS” such as *supermarkets*, *flea markets*, and *stalls* are where users buy commodities. “Eating AS” such as a *dining room* at home, *restaurants*, and *cafeterias* are where we eat and drink. “User’s own domains AS” such as *the user’s own room* in their house and *the user’s desk* at the office are where the user keeps his/her possessions. Activity spaces are not simply spaces defined in terms of X-Y-Z coordinates with no regard for semantics; activity spaces are inherently associated with semantics. Activity space is a subconcept of place. With regard to place, Tuan [18] mentioned that “place is space infused with human meaning”, and Curry [1] mentioned the several ways in which places are created: naming, categorizing, making a symbol, telling stories, and performing activities. In his categorization, an activity space is a place of performing activities with particular objects.

Activity spaces have the following characteristics.

**Dynamics of existence:** Activity spaces are dynamically generated, move, and disappear. For example, “a shopping AS” such as a *flea market* is dynamically generated, moves, and disappears in parks or squares depending on the action of the booth owners. “An eating AS” can be dynamically generated by preparing a meal and disappears after the meal. Each activity space has a different period of existence. Some activity spaces, such as a bedroom in a house, can exist for long periods. On the other hand, the activity spaces such as an eating activity space or a flea market exist for short periods. This characteristic raises a technical issue: the transient activity spaces cannot be identified by using spatial maps.

**Spatial relationships:** Several Activity Spaces can exist at the same spatial position. For example, while a living room is designed to enable people to get together for meeting or chatting with a family or friends, people do several other activities such as eating and working in a living room. Therefore, there are spatial relationships among activity spaces such as inclusion, overlap, and adjacency. This characteristic raises the fact that multiple activity spaces can occupy the same spatial position.

Therefore, a key technical issue on activity space is multiple activity space detection.

#### 3.2 Thing-Oriented Activity Spaces Detection and Its Difficulties

People can generally recognize an activity space simply by “looking at” it. For example, if people look at a kitchen in a house, they can recognize it as a *cooking AS*. The

reason is affordance as introduced by Gibson [4]. Affordance is what things offer people to do with them. Affordance enables you to recognize what actions you can do with a thing by just looking at it. For example, *a knife* offers the function of *cutting* to people and they can recognize that *a knife* can be used to *cut* objects by just looking at *the knife*. By extending affordance, we believe that a set of things also affords particular activities to people. Therefore, we focus on sets of things to identify activity spaces. However, identifying activity spaces from a set of things suffers from several difficulties. We listed them based on the characteristics of things.

**Massiveness:** People are surrounded by a huge number of things. While some of them are effective in identifying the activity space, others such as lamp and trash are useless; eliminating the ineffective things is very difficult [P1-1].

**Mobility:** Things can be moved by several causes. The things that are moved due to the user's intention such as food or dishes for preparing meals are important in identifying the activity space. However, other things that move such as the user's clothing are useless; it is necessary to suppress the noise [P1-2].

**Polysemy:** Everything has multiple representations. For example, the thing *pencil* has the meaning of *a writing tool* and *stationery*, and at the store, it has the meaning of *a commercial goods*. Thus, how to represent things is difficult [P2].

**Variability:** The things that form the same kind of the activity space are different in each activity space. For example, *the cooking ASs* of different houses have different things. This implies the manual creation of detection rules is extremely difficult [P3]. Furthermore, even if some learning approaches are utilized to automatically extract inference rules, the system cannot deal with unlearned things [P4].

While the above difficulties arise from the characteristics of things, other problems arise from the use of RFIDs: RFID tag detection is not completely reliable because of collision and differences in the interval of ID transmission.

### 3.3 Ontology and PMM for Detecting Activity Spaces from Things

To solve P2 and P4, we utilize ontology that defines explicit formal specifications of terms and the relations among them. As for P1-1, P1-2, P3, and the technical issue of multiple activity spaces detection, we employ the parametric mixture model (PMM) [20], a text classification method, because we draw an analogy between documents composed by words and activity spaces composed by things.

The proposed system consists of four processes: preprocess, represent, learn, and classify. In the preprocess stage, the system aggregates detected RFID tags and extracts distinct things. For example, the system extracts only things that appeared recently to detect newly generated activity spaces. In the represent stage, the system acquires terms that represent each thing. We acquire the attribute information of each detected thing from Physical Markup Language servers (PML servers) [12] of EPC Global. Utilizing the information, all terms representing the things are acquired through ontology. At the learn stage, the probability of a thing being in an activity space is specified by utilizing the terms and supervised activity space data. In the classify stage, the system uses PMM to classify a set of terms into activity spaces.

### 3.3.1 Ontology to Manage Representations

Ontology has a long history in philosophy as it refers to the subject of existence. One definition of ontology involves the specification of terms in each domain and the relations among them. Ontology sets “basic concept” that represent underlying concept such as *pencil* and “role concept” that represent the role that a thing plays in a particular domain such as *product*. In addition, it also sets “is-a relation” to represent the sub concept between two terms. For example, “A pen *is-a* writing tool” means a *pen* is a sub concept of a *writing tool* [8][9][10]. Utilizing these concepts and relations makes it possible to acquire all terms related to a thing by tracing relations. The lowest terms in each concept are preliminarily linked to the ID of each thing in PML. Among all terms related to a thing, it is necessary to identify the appropriate term for the thing to identify activity spaces. Since it is difficult to identify proper terms and the appropriate level in a hierarchy based on *is-a* relation, we manually choose the appropriate concept instead of the appropriate terms. For example, *basic concept* is selected for a *working AS*, and *role concept* is selected for a *shopping AS*. This approach, however, leaves unanswered how to select the proper abstraction level in the *is-a* relation; this is solved in the next section. To solve P4, we transform the terms that have not been learned into the terms that have been learned by utilizing *is-a* relations. For example, if the thing *eraser* has not been learned but the thing *pencil* has, we can treat both as *stationery*, which has already been learned.

### 3.3.2 Activity Space Identification Via Topic Detection

Many schemes for tackling the identification of the topics of documents or web contents have been proposed. The characteristics of their target are very similar to those of our objective: a document consists of a set of words that includes noise such as stop words [15], each document on the same topic consists of different words, but people can identify the topic of a document at a glance. Among the many approaches proposed for topic detection, most assume that a document has only one topic; the parametric mixture model (PMM), however, allows one document to have multiple topics. It employs a probabilistic approach which is efficient and robust against noise; it offers the highest accuracy in detecting multiple topics [20]. Since it is highly likely that multiple activity spaces will be detected from one set of things, we employ PMM.

PMM extends Naïve Bayes [6] to provide multi-topic detection. PMM assumes that a multi-topic document is composed of a mixture of words typical of each topic. Based on this assumption, a multi-topic document can be represented as the linear summation of the word occurrence probability vector of each topic as shown in Eq. (1). Here,  $p(t_i|c_i)$  is calculated using MAP estimation. By replacing (words, topics) with (things, activity spaces), we can use Eq. (1) to detect multiple activity spaces.

$$p(d|c) = p(t_1, \dots, t_n|c) = \prod_{i=1}^n \left( \sum_{l=1}^L h_l(y) p(t_i|c_l) \right)^{x_i}$$

$$\text{where } h_l(y) = \frac{y_l}{\sum_{l=1}^L y_l}, l = 1, \dots, L, y_l = 1 (y_l \text{ belongs}) \text{ or } 0 (y_l \text{ does not belong}) \quad (1)$$

$d$  : document,  $c$  : topic,  $x_i$  : frequency of word  $t_i$ ,  $L$  : No. of topic,  $n$  : No. of word kinds

To select the appropriate abstraction level of an *is-a* relation, conditional probability  $p(\text{thing} | \text{activity space})$  is learned utilizing the lists of things at each abstraction level.



PMM then acquires the classification accuracy of activity spaces though the learned conditional probability. Finally, the abstraction level with the highest classification accuracy is employed to classify a test set of things.

## 4 Preliminary Experiment

Before doing experiments in an actual environment, we did preliminary experiments using actual data that was manually collected. We evaluate the feasibility of the proposed method for activity space detection under the difficulties posed by P1-1, P2, P3, and P4 while P1-2 and the difficulties created by the RFIDs was left for the experiments in actual environments. We did two experiments: exp.1), the detection of frequently changing activity spaces to address P2, P3, and P4, and exp.2), the detection of an activity space that contains a large number of things to address P1-1. In exp.1, we focused on a table in a living room since it can support several activities as described in 3.1. Since a meeting AS always exists, we detect three activity spaces: just a meeting AS, a meeting AS and a working AS, and a meeting AS and an eating AS. We assume that RFID tag readers are put on the table and detect things on or near the table. Though activity spaces on a table in a living room frequently change, the things on it are relatively few (94 things, 26 kinds). In exp.2 we focused on a room in a home since each room has many things (836 things, 472 kinds). We detect four types of activity spaces: a living room (a meeting AS), a kitchen (a cooking AS), a bath room (a bathing AS), and a study room (a working AS). We assume that each room has several RFID tag readers. We use F-measure to evaluate the accuracy of activity space detection. The F-measure is defined as the harmonic mean of precision and recall and is widely used in the information retrieval field.

### 4.1 Input Data to Detect Activity Spaces

First of all, we need to acquire the data of real world things that includes actual noise. As for exp.1, we manually identified all things on a table of a typical Japanese home. Although PMM must know of the things of each activity space to learn the conditional probability, a meeting AS always exists when a working AS or a meeting AS exists. Therefore, we eliminate the data of things that indicate just a meeting AS from those of an eating AS or a working AS. Fig. 1 (a) shows the things of each AS. As for

(a)	Activity Space	Things	
	Living place	1 dining table, 4 chairs, 4 cushions, 4 newspapers, 1 vase, 1 jotter, 5 window envelops, 5 ballpoints, 1 in-basket, 1 wastepaper basket, 2 coasters	
	Eating place (Breakfast)	6 dishes, 2 chopsticks, 2 table spoons, 2 mugs, 2 table linens	
	Eating place (Lunch)	6 dishes, 2 chopsticks, 2 forks, 2 table knives, 2 glasses, 2 table linens	
	Eating place (Dinner)	6 dishes, 2 chopsticks, 2 forks, 2 table knives, 2 glasses, 2 beer cans, 2 table linens	
	Working place	2 ballpoints, 4 highlighters, 1 commonplace book, 1 digital computer, 1 power code, 1 mouse, 7 files	

**Fig. 1.** (a): Things of each activity space for exp.1. (b), (c): Pictures used to identify things in an actual office desk for exp.2.

exp.2, we used the things in an actual Korean family's house as collected by the National Museum of Ethnology [16]. Since the house did not have a study room, we manually identified all things in and on an office desk from photos taken at various angles (Fig. 1 (b) (c))

To represent the data of things in abstract terms, we surveyed existing ontology bases in terms of the number of vocabularies, abstract terms, and hierarchy and the structure of conceptualization. In this survey, we decided to employ WordNet [23]. We set "artifact" in WordNet as abstraction level 1, the most abstract term, and acquired terms on abstraction levels 2 to 6 by utilizing the *is-a* relations provided by WordNet. Instead of utilizing PML, we manually set the terms of abstraction level 6 representing each thing. We then added noise to the abstraction data sets with noise ratios of 0%, 25%, and 50%. In detail, we added the things of another activity space to reflect the presence of things not related to the activity space. In addition, we randomly eliminated some things from the data sets to reflect RFID detection error and the presence of things without an RFID tag. By randomly adding noise, we created 1000 data sets for each activity space. To include unlearned things in test data for evaluating P4, we used eating ASs (breakfast) as learning data and those of lunch and dinner as test data in exp.1.

## 4.2 Results

Table 1 shows the F-measure in exp.1. The proposed method successfully detected a meeting AS and an eating AS with a high degree of accuracy, while that of a working AS was not high. This result indicates that a meeting AS and an eating AS have the particular things that clearly identify the activity space while a working AS does not. In detail, while the multiple activity spaces of working and living can be successfully detected, the single activity space of living is classified as the multiple activity spaces of working and living. This is because a meeting AS has some characteristic things of such as ballpoints and jotters. This result also demonstrates the noise tolerance of the proposed method since the accuracy of activity space detection did not drop as the noise ratio was raised. Furthermore, the accuracy of working AS detection increased with the abstraction level. By raising the abstraction level, the number of kinds of terms decreased: 1 kind in level 1 and 34 kinds in level 6. This means that the information amount decreased and the accuracy of activity space detection generally falls. Ontology can provide an explanation: each activity space has many kinds but a few discriminative terms; the use of ontology raised the abstraction level which transformed them into fewer kinds with a larger number of discriminative terms. This demonstrates that ontology can raise the accuracy of activity space detection. Note that it makes sense that the F-measure of level 1 is 0 in most activity spaces since the term of abstraction level 1 is just "Artifact". As for unlearned things, we did not learn *forks*, *table knives*, and *glasses*. WordNet transformed *forks*, *table knives*, and *tablespoons* into *cutlery* in level 5. *Glasses* and *mugs*, which were known were transformed into *container* in level 3. Therefore, ontology could utilize unlearned things for activity space detection by raising the abstraction level.

Table 3 shows the F-measure of each activity space in exp.2; the results also demonstrate the feasibility of the proposed method. Table 2 shows the processing time needed for assessing 4000 sets of things data and the number of kinds of terms in each abstraction level. This demonstrates that the proposed method can rapidly handle

large sets of things and that increasing the abstraction level makes it possible to reduce the processing time. Furthermore, though 472 kinds of things were aggregated into 17 kinds in abstraction level 2, the F-measure of each activity space did not drop, which obviously demonstrates the effectiveness of ontology.

**Table 1.** F-measure of detecting each activity space in exp.1

Meeting AS						
noise ratio	Level1	Level2	Level3	Level4	Level5	Level6
0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
25%	100.0%	99.9%	100.0%	100.0%	100.0%	100.0%
50%	100.0%	98.3%	100.0%	100.0%	100.0%	100.0%

Meeting AS and Eating AS						
noise ratio	Level1	Level2	Level3	Level4	Level5	Level6
0%	0.0%	100.0%	100.0%	100.0%	100.0%	100.0%
25%	0.0%	100.0%	100.0%	100.0%	100.0%	100.0%
50%	0.0%	100.0%	100.0%	100.0%	99.8%	100.0%

Meeting AS and Working AS						
noise ratio	Level1	Level2	Level3	Level4	Level5	Level6
0%	0.0%	100.0%	100.0%	100.0%	100.0%	100.0%
25%	0.0%	75.9%	93.5%	94.2%	96.4%	90.5%
50%	0.0%	62.9%	81.0%	79.3%	86.1%	78.0%

**Table 2.** Processing time for estimating 4000 data sets and the No. of kinds of terms

	level1	level2	level3	level4	level5	level6
No. of kinds of terms	1	17	48	149	421	472
time for estimation (sec)	37	55	100	227	526	570

## 5 Discussion

The preliminary experiments described above evaluated the proposed method using actual but manually collected sets of things. Though the findings of these experiments are meaningful and interesting, some problems remain to be evaluated.

**Mobility of things:** Noise data derived from this characteristic are not included in the manually collected sets. To evaluate it, we need to establish an environment where each thing has an RFID tag and gather the data of RFID tag detection over time.

**Human activity inference based on activity spaces:** We need to evaluate the accuracy of inferring the user's situation from the activity spaces. To do this, we can compare the activity inferred from activity spaces with actual user activity acquired by asking the user what s/he is doing in the environment.

Furthermore, we need to consider the following issues.

**Target activity spaces:** Activity spaces need to be expanded and refined. As for expansion, we need to acquire as many activity spaces as there are human activities. As for refinement, we need to classify each activity space into more refined activity space. For example, a *meeting AS* has sub-concepts of a *director's meeting AS* and a *group meeting AS*. Ontology would be helpful in achieving this.

**Table 3.** F-measure of detecting each activity space in exp.2

Bathing AS						
noise ratio	Level1	Level2	Level3	Level4	Level5	Level6
0%	0%	100.0%	100.0%	100.0%	100.0%	100.0%
25%	0%	97.1%	95.6%	99.8%	98.9%	100.0%
50%	0%	86.6%	92.2%	99.4%	97.8%	100.0%

Cooking AS						
noise ratio	Level1	Level2	Level3	Level4	Level5	Level6
0%	40%	100.0%	100.0%	100.0%	100.0%	100.0%
25%	40%	91.1%	96.5%	98.7%	99.4%	100.0%
50%	40%	82.5%	92.5%	97.6%	96.9%	100.0%

Meeting AS						
noise ratio	Level1	Level2	Level3	Level4	Level5	Level6
0%	100%	100.0%	100.0%	100.0%	100.0%	100.0%
25%	100%	94.6%	97.2%	98.1%	98.8%	100.0%
50%	100%	89.9%	94.0%	97.5%	97.7%	100.0%

Working AS						
noise ratio	Level1	Level2	Level3	Level4	Level5	Level6
0%	0%	100.0%	100.0%	100.0%	100.0%	100.0%
25%	0%	90.6%	97.5%	99.3%	99.4%	100.0%
50%	0%	83.9%	95.1%	98.4%	97.4%	100.0%



**Concepts of ontology:** WordNet defines only the basic concepts of terms. Though it is useful to identify many activity spaces, other activity spaces such as *selling space* are difficult to identify. Since no existing ontology base defines role concept such as *product*, we need to build the ontology of role concept.

**PMM for activity space detection:** Since PMM is designed for identifying a topic from a large amount of words, it is difficult to identify the few important things. For example, *takeout food* consists of few things but important in identifying *an eating AS*. This issue can be solved by adding weight to thing in preprocessing.

**Human activity inference in multiple activity spaces:** When multiple activity spaces overlap, the system needs to choose some of them to identify user activity. One approach is identifying the activity space generated most recently since a user intentionally moves things to perform a particular action such as preparing meal. Another approach is identifying the thing that the user is interacting with and specifying the activity space with the highest conditional probability.

## 6 Conclusion

This paper proposed a novel approach to recognize human activities based on activity spaces, the spaces that afford humans particular activities. Activity spaces are identified through the things that exist there based on the concept of affordance. We utilize ontology to specify terms representing things and the parametric mixture model to identify activity spaces from the terms. Since activity spaces represent the “actual” semantics of position, activity spaces infer human activities more precisely than conventional approaches based on just location; moreover, this approach is more feasible than those based on just what the user is interacting with. Preliminary experiments demonstrated the noise tolerance, high accuracy of activity space detection, and the ability to rapidly handle large amounts of data. Though we focused on human activities with things, other activities that are independent of things remain to be recognized. Such activities may depend on other entities such as human or time context and in that case, our approach based on ontology and topic detection may be applicable.

## References

1. Curry M.: *The Work in the World – Geographical Practice and the Written Word*. University of Minnesota Press, ISBN 0-8166-2665-0, 1996.
2. EPC Global: <http://www.epcglobalinc.org/>
3. Fishkin K., Jiang B., Philipose M., Roy S.: *I Sense a Disturbance in the Force: Unobtrusive Detection of Interactions with RFID-tagged Objects*. Proc of 6th Intl. Conference on Ubiquitous Computing (UbiComp2004), pp.268-282, 2004.
4. Gibson J.: *The Ecological Approach to Visual Perception*. Lawrence Erlbaum Assoc Inc, ISBN: 0898599598, 1979.
5. Gruber T.: *A translation approach to portable ontologies*. Knowledge Acquisition, 5(2):199-220, 1993.
6. McCallum A., Nigam K.: *A Comparison of Event Models for Naïve Bayes Text Classification*. Proc. of Intl. Workshop on Learning for Text Categorization in AAAI-98, 1998.

7. Moore D., Essa I., Hayes M.: Exploiting Human Actions and Object Context for Recognition Tasks. Proc. of 4th Intl. Conference on Computer Vision (ICCV'99), 1999.
8. Mizoguchi R.: Tutorial on ontological engineering Part1: Introduction to Ontological Engineering, New Generation Computing, OhmSha&Springer, Vol.21, No.4, pp.365-384, 2003.
9. Mizoguchi R.: Tutorial on ontological engineering Part2: Ontology development, tools and languages, New Generation Computing, OhmSha&Springer, Vol.22, No.1, pp.61-96, 2004.
10. Mizoguchi R.: Tutorial on ontological engineering Part3: Advanced course of ontological engineering, New Generation Computing, OhmSha&Springer, Vol.22, No.2, 2004.
11. Nishida Y., Kitamura K., Hori T., Nishitani A., Kanade T., Mizoguchi H.: Quick Realization of Function for Detecting Human Activity Events by Ultrasonic 3D Tag and Stereo Vision. Proc. of 2nd IEEE Intl. Conference on Pervasive Computing and Communications (PerCom2004), pp. 43-54, 2004.
12. PML Core Specification 1.0: [http://www.epcglobalinc.org/standards\\_technology/Secure/v1.0/PML\\_Core\\_Specification\\_v1.0.pdf](http://www.epcglobalinc.org/standards_technology/Secure/v1.0/PML_Core_Specification_v1.0.pdf)
13. Schilit B., Adams N., Gold R., Tso M., Want R.: ParcTab Mobile Computing System. Proc. of 4th Workshop on Workstation Operating Systems (WWOS-IV), pp.34-39, 1993.
14. Seon-Woo L., Mase K.: Activity and Location Recognition Using Wearable Sensors. Pervasive Computing, pp.10-18, Sep.2002.
15. Stop list: <ftp://ftp.cs.cornell.edu/pub/smart/english.stop>
16. The National Museum of Ethnology: Seoul Style 2002. ISBN: 4915696465, 2002.
17. Tapia E., Intille S., Larson K.: Activity Recognition in the Home Using Simple and Ubiquitous Sensors. Proc. of 2nd Intl. Conference on Pervasive Computing 2004 (Pervasive2004), pp.158-175, 2004.
18. Tuan Y.: Space and Place: The Perspective of Experience. ISBN: 0816638772, 1977.
19. Ubiquitous ID Center: <http://www.uidcenter.org/>
20. Ueda N., Saito K.: Singleshot detection of multi-category text using parametric mixture models. Proc. of 8th Intl. Conference on Knowledge Discovery and Data Mining (SIGKDD2002), pp. 626-631, 2002.
21. Ward A., Jones A., Hopper A.: A New Location Technique for the Active Office. IEEE Personal Communications, Vol. 4, No. 5, pp.42-47, 1997.
22. Weiser M.: The Computer for the 21st century. Scientific American, pp.94-104, Sep.1991.
23. WordNet: <http://wordnet.princeton.edu/>

# Baton: A Service Management System for Coordinating Smart Things in Smart Spaces

Jingyu Li and Yuanchun Shi

Key Laboratory of Pervasive Computing, Ministry of Education,  
Department of Computer Science and Technology,  
Tsinghua University, Beijing 100084, China  
lijingyu03@mails.tsinghua.edu.cn, shiyc@tsinghua.edu.cn

**Abstract.** Smart spaces are open complex computing systems, consisting of a large variety of cooperative smart things. Central to building smart spaces is the support for sophisticated coordination among diverse smart things collaborating to accomplish specified tasks. Multi-agent systems are often used as the software infrastructures to address the coordination in smart spaces. However, since agents in smart spaces are dynamic, resource-bounded and have complicated service dependencies, current approaches to coordination in multi-agent systems encounter new challenges when applied in smart spaces. In this paper, we present Baton, a service management system to explicitly resolve the particular issues stemming from smart spaces while coordinating agents (delegating smart things in smart spaces). Baton is designed as a complement to coordination approaches in multi-agent systems with a focus on mechanisms for service discovery, service composition, request arbitration and dependency maintenance. Baton has been now deployed in our own smart spaces to achieve better coordination for smart things.

## 1 Introduction

Smart Spaces [1] are typically open, distributed, heterogeneous and dynamic computing systems, which can be conceived as cooperating ensembles of a great variety of smart things, striving to achieve different missions. Accordingly, when building smart spaces, a fundamental aspect should be to support sophisticated coordination among diverse smart things working together towards accomplishing specified tasks. Many research projects have adopted multi-agent systems to enforce the coordination in smart spaces [2][3][4], where smart things are delegated by Agents, who provide services and consume services, and are coordinated in terms of service dependencies. However, existing coordination mechanisms of multi-agent systems need to be enhanced to cope with the particular situations when coordinating smart things in smart spaces [5].

1) Smart spaces are open and dynamic environments, where smart things (i.e. a smart phone) may enter or leave at will. Along with the smart things' appearance or absence, services provided by them are dynamically available or disappearing in a smart space, making the service consumers experience discontinuous services and thus tampering with the consumers' tasks. As a result, coordination in smart spaces

needs to continuously maintain the service dependencies among smart things in spite of variations of service providers.

- 2) Smart things in smart spaces are resource-bounded since they are integrated with physical environments where physical resources are limited in number and have many physical constraints. So when there are more requests than a service provider can handle, for example, a video player, an email notifier and a file controller simultaneously require to use the only wall-sized display of a smart space, service request collisions will be incurred. Therefore, coordination in smart spaces should try to resolve request collisions and ensure that each consumer can get its deserved services so as to satisfy its service dependencies.
- 3) Besides the simple case in which a requested service can be directly provided by a single smart thing, smart spaces often encounter more complex situations where the requested service has to be fulfilled through the orchestration of multiple smart things conforming to certain control logic. A typical scenario is that a user may submit a PPT-Display service requirement, which must be satisfied by cooperating at least a File Reader and a Projector, or more considerably, a lamp controller (to dim the light for better vision), a laser pointer (to control the PPT files) and so on. Consequently, to form new high-level services, coordination in smart spaces needs to comply with some coordination rules in addition to simple service dependencies.

Some of the existing multi-agent systems [13] perform agent coordination by means of high-level agent communication language and conversation protocols, such as FIPA ACL and KQML, which assumes that the interaction patterns are established in a priori and thus doesn't appropriately support the notion of openness and dynamicity [6]. OAA [2] provides a loose-coupling framework to accommodate dynamic agents and utilizes a "delegating computing" notion to coordinate agents, but doesn't deeply consider the problem of service request collisions. Metagluce [3] enforces its agent coordination with the assistance of a dedicated resource management system, called Rascal. Rascal [7] deals with many of the issues pertinent to smart spaces when coordinating agents, such as resolving request collisions, however, it doesn't take many considerations on composing several services to fulfill a potential request [12], which are common cases in smart spaces and doesn't pay much attention to maintaining service dependencies when agents join or leave smart spaces.

In this paper, we present Baton, a service management system to explicitly address the particular issues in smart spaces. Baton can be regarded as a complement to coordination approaches of multi-agent systems with a focus on mechanisms for service discovery, service composition, request arbitration and dependency maintenance. Baton is implemented on Smart Platform [4], which is a multi-agent system designed as the software infrastructure for our own smart spaces (Smart Classroom [8] and Smart Meeting Room [9]), where agents are loose-coupled and interact in a black-board pattern.

The remainder of this paper is organized as follows: section 2 first defines some basic concepts, then presents an overview of the structure of Baton, and gives a detailed description of its two major components, *Knowledge Base* and *Service Broker*, which play key roles in addressing the particular issues when coordinating agents in smart spaces. Section 3 concludes the whole paper and discusses our future work.

## 2 Architecture of Baton

### 2.1 Basic Concepts

Before detailed introduction of Baton, we'd like to clarify several concepts relating to our work.

1. **Agents.** Agents are basic functional units in smart spaces, who provide services and in the meanwhile, consume services. Note that all the smart things are encapsulated as agents in our smart spaces. We assume that agents in smart spaces are trusted and friendly -- they can honestly express their service needs and capabilities, and release the services when they have finished their jobs or when some others need those services badly.
2. **Services.** Services are well defined functionalities provided by smart things, delegating by agents. An agent can provide one or more services, and a service may be provided by a number of agents. For example, a smart mirror may provide face recognition service and experience capture service, while a controlling service can be provided by a laser pointer or a speech recognizer. Services are the interfaces through which agents interact and cooperate with one another.
3. **Atomic Service and Composite Service.** An atomic service can be provided by a single agent, while a composite service has to be fulfilled through the teamwork of multiple agents. For example, in an automated office [14], a message notification service needs to be accomplished by at least a message receiving agent, a user location detection agent and a text or speech output agent.

### 2.2 Structure

The architecture of Baton is shown in Figure 1. Each of the smart spaces needs to run an instance of this architecture so as to perform its service management.

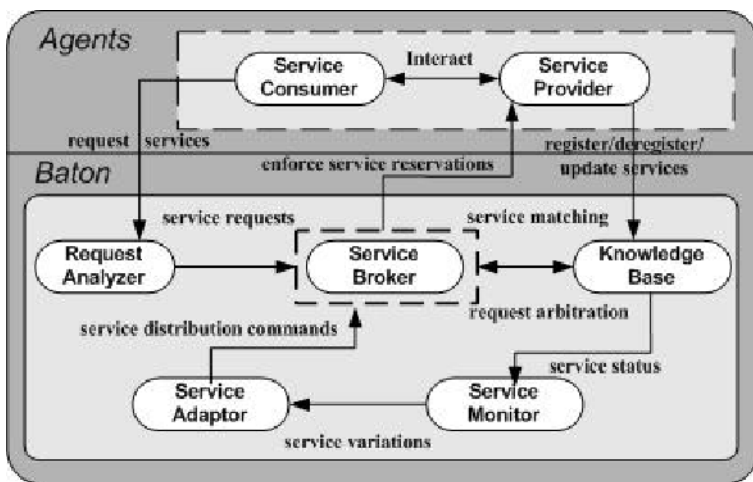


Fig. 1. Architecture of Baton

Baton consists of five components, which are *Knowledge Base*, *Service Broker*, *Request Analyzer*, *Service Monitor* and *Service Adaptor*. For the sake of efficiency, each of these components is encapsulated as a single agent, distributed among different devices and cooperating to fulfill the tasks of Baton.

*Knowledge Base* stores all the information about the services registered by agents, and performs the service-request matching. *Service Broker* is the core part of Baton, and takes charge of choosing the most suited services, deciding who should get the requested service when service request collision happens and constructing composite services. In fact, *Service Broker* (enveloped by dashed lines) may be extended to several distributed federated cooperating *Service Brokers* when the smart spaces become larger or there are more services. *Request Analyzer* translates the service requests into internal representations that can be parsed by *Service Broker*. *Service Adaptor* here is responsible for maintaining the service dependencies among agents in case of changes, and mainly considers two situations: (1) if a previous service request hasn't been satisfied, then once the desired service appears in the system, *Service Adaptor* will help to establish this service dependency; (2) if a service consumer loses its service while it is being served, *Service Adaptor* will try to find another substitute to continue this service. *Service Monitor* monitors and collects the service information in a real-time fashion. We are now extending *Service Monitor* from a component in Baton to a visualized tool, through which we can see clearly what kinds of services are active in a smart space, what they can do, and what status they are in.

In the following sections, we will give a further discussion on the two major components of Baton, the *Knowledge Base* and the *Service Broker*.

### 2.3 The Knowledge Base

*Knowledge Base* contains the information of both atomic services and composite services, and thus is the basis on which Baton makes all its decisions when coordinating agents. Atomic service information comes from the service descriptions submitted by agents when they first participate in a smart space, and is updated by agents themselves when change occurs in their lives. While composite service information is the knowledge about how to construct a composite service, which may include what atomic services are needed and what their logical relationship is to form this composite service.

**Service Descriptions.** We have recognized that the descriptions of services should mainly cover the following two aspects:

- 1) **Inherent information.** Inherent information describes the inherent features of a service, including service name, attributes and values, maximum capacity, provider, and service dependencies, which specify what other services are needed to provide this service, for example, the Speaker Recognition agent in Smart Classroom [8] often depends on the aid of the User Profile agent to correctly recognize the speaker. Inherent information also includes the interface information, which specifies how the service should be accessed and interacted with. With the purpose of achieving the automatic service invocation and interoperation, we utilize OWL-S [15] as the description language to describe the inherent information of a service.
- 2) **Dynamic information.** Dynamic information reflects the runtime states of a service, which describes to what extent the service has been used and how many of its

service dependencies have been satisfied. A service can be *free*, *reserved*, *busy* or *busy reserved*, its dependencies can be *satisfied*, *unsatisfied* or *satisfying*, and its current available capacity varies with its workload. Dynamic information is initialized as *free*, *unsatisfied* and *maximum capacity*, and is dynamically updated by the service provider in case of changes. Since OWL-S doesn't take many considerations on the runtime status of a service, we use XML language to describe the dynamic information to supplement the description of a service. Baton keeps track of the services by examining the descriptions of their dynamic information, which reveal every detail of the services in their whole lives.

**Knowledge Representation for Composite Services.** To accurately express the complicated relations among atomic services cooperating to perform different composite services, we borrow the idea of ConcurTaskTrees [10] to model a composite service. The tree-like structure with relational operators proposed by ConcurTaskTrees guarantees the integrity and clarity of the specifications of a wide variety of composite services. The nodes and relational operators used in the model of our composite service trees are defined as follows:

- 1) **CS.** CS delegates a composite service to be constructed, which can be a root node or an internal node of a composite service tree.
- 2) **AS.** AS is an atomic service, always being a leaf node of a composite service tree.
- 3) The operators describing the temporal relationships of the services are only applied to those on the same level of a composite service tree. In view of the current smart spaces, we only define four operators:
  - **S1 >> S2:** Service S2 is activated when S1 terminates. For example, in Figure 2(a), as with a PPT-Display service, the projecting service only makes sense after the File-access service finishes.
  - **S1 []>> S2:** When service S1 terminates, it provides some values for S2 besides activating it. A typical example is shown in Figure 2(b), a Bitmap-to-PPT transformation service can be performed by a Bitmap-to-x service first, and then an x-to-PPT service. Here x can be any transitional format, such as gif format.
  - **S1 | S2:** choosing. As is shown in Figure 2(a), PPT-controlling can be performed by hand-free controlling like speech command, or manual controlling like pointing.
  - **[S]:** S is optional. In Figure 2(a), marking service is optional and is activated only if the consumer needs to make annotations on the PPT file.

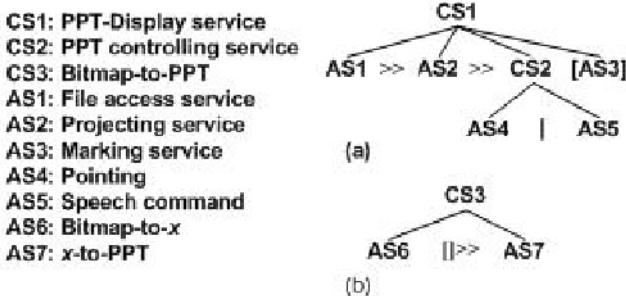


Fig. 2. Examples of composite service trees

The knowledge for a composite service may either be well-predefined by application designers, who just need to specify the composition strategy by using the ConcurTaskTrees model (e.g. Fig.2.(a)), or be generated based on the OWL ontology of the atomic services (e.g. Fig.2.(b)).

In terms of our own smart spaces [8] [9], a number of possible composite services have been modeled using the above nodes and operators. And for consistency, the ConcurTaskTrees models are also mapped to OWL-S descriptions. For convenience and efficiency, the Berkeley DB XML [11], a dedicated database for mastering xml files is introduced in Baton as the *Knowledge Base* to maintain the descriptions of all services, and a query language, Xpath [16] is utilized and extended to access the database.

## 2.4 The Service Broker

**Evaluation for Service Matching.** In spite that there may be several candidate services for a request found in *Knowledge Base*, it is probable that some of them may be insufficient and some may be an oversupply. For example, as to a request for a color printing service, if a color printer doesn't exist, a black-and-white one can be a substitute but is inadequate; while as to a request for a video playing service with maximum frame rate of 30fps, the service offering 40fps will be a waste. Actually, there exists an affinity between a service and a request --- the more perfectly the service matches the request, the closer the affinity. Given a request  $R$ , a candidate service  $S$ , then  $Affinity(R, S) = Match(R, S)$ , where the value of  $Match(R, S)$  is estimated according to several predefined matching rules,  $Match(R, S) \in [0, 1]$ .

In simple cases, *Service Broker* just picks the free service owning the highest *Affinity* value above a preset threshold as the final choice for a request. However, if all the candidate services are being occupied, the process of solving request collisions will be activated.

**Solution for Request Collisions.** Services provided by agents in smart spaces are capacity-limited, so if multiple service consumers request the same service, a verdict must be made on who should get the service. While solving request collisions, we recognize that three guidelines should be followed in smart spaces:

- 1) Agents have different priorities in smart spaces, for example, an agent delegates a teacher in Smart Classroom has a much higher priority than those delegating students. So when different agents contend for the same service, priority is a very important factor in deciding to whom the service should go.
- 2) Agents in smart spaces tend to be served continuously, rather than frequently disturbed. So if request collision occurs and service redistribution is inevitable, then the changes of service dependencies brought by the redistribution should be minimized.
- 3) Service types in smart spaces are varied, so it is difficult to achieve a global optimal distribution of all services, but reaching an optimal distribution of a single service is quite feasible.

According to these guidelines, we take subsequent considerations:

- Service consumers usually have contentment evaluations on the results of their service requests. Specifically, a contentment evaluation basically relies on whether the consumer can get its desired service, and how the affinity between the request



and the service is like. A formal description of the contentment evaluation is: Given a request  $R$ , a service  $S$ , then

$$\text{Contentment}(R, S) = \text{Affinity}(R, S) \times \text{Available}(R, S) \quad (1)$$

$\text{Available}(R, S)$  demonstrates that whether service  $S$  can be used by  $R$ , if yes, then  $\text{Available}(R, S) = 1$ , otherwise,  $\text{Available}(R, S) = 0$ .

- However, since *Service Broker* has to dynamically adjust the distributions of services in response to requests submitted by agents at will, consumers' contentment evaluations on a certain service vary from time to time. To be specific, for example, when the  $N$ th request on service  $S$  comes, consumer  $A$ 's contentment evaluation on  $S$  may be 0, as its request for  $S$  hasn't been fulfilled; whereas, when the  $N+1$ th request on  $S$  comes,  $A$ 's contentment evaluation may increase to 1 because it has acquired  $S$  for some reason, and when the  $N+2$ th request comes,  $S$  may be taken away from  $A$ , making  $A$  quite displeased, and  $A$ 's contentment evaluation may decrease to -1. To reflect that contentment evaluation is changing with new coming request, we formulate contentment evaluation with a variation of equation (1):

$$\text{Contentment}(R, S, N) = \text{Affinity}(R, S) \times f(R, S, N) \quad (2)$$

$N$  indicates the  $N$ th request on  $S$  and  $f(R, S, N)$  is defined as follows:

$$f(R, S, N) = \begin{cases} 1 & \text{if } \text{Available}(R, S, N) = 1 \\ 0 & \text{if } \text{Available}(R, S, N-1) = \text{Available}(R, S, N) = 0 \\ -1 & \text{if } \text{Available}(R, S, N-1) = 1 \ \& \ \text{Available}(R, S, N) = 0 \end{cases} \quad (3)$$

- The goal of *Service Broker* in solving request collisions is to take every effort to fulfill each request for a certain service so as to maximize the total contentment evaluations of all the consumers on this service, and ensure that this service is not exploited beyond its capacity. Therefore, the problem of solving request collisions turns out to be a constraint satisfaction problem. It is reasonable to believe that when two consumers request the same service, distributing the service to the one who has a higher priority will take more benefits to the sum of contentment evaluations, so the priority value of each consumer can be assigned as the weight of its contentment evaluation when calculating the totals. Consequently, as with a given service  $S$ , when the  $N$ th request on  $S$  comes and incurs a collision, the goal of *Service Broker* can be clearly illustrated as:

$$\begin{aligned} \text{Max } C(S, N) &= \sum [\text{Priority}(R_i) \times \text{Contentment}(R_i, S, N)] \\ \text{s.t. } \sum [\text{Require}(R_i, S) \times \text{Available}(R_i, S, N)] &\leq \text{Maximum Capacity}(S); \\ R_i \in RA &= \{\text{Requests on } S \text{ from service consumers}\}; \\ \text{Available}(R_i, S, N) &\in \{0, 1\}; \\ i &= 1, 2 \dots |RA|; \end{aligned} \quad (4)$$

Therein,  $\text{Require}(R_i, S)$  specifies the requirement that  $R_i$  poses on  $S$ , which will be discussed later.

- In terms of capacity, we identify two distinct categories of services:
  1. Capacity of a service means the largest number of consumers that the service can support in parallel. For example, the capacity of a speech recognition agent may be that it can simultaneously handle three channels of speech stream. As to this case,

any consumer can only get a copy of the service, thus  $Require(R_i, S)=1$ , and the constraint condition in (4) can be simplified to:

$$\Sigma Available(R_i, S, N) \leq Maximum\ Capacity(S) = 3.$$

2. Capacity of a service may have no explicit confinement on the number of consumers, but is limited by its own capability. For example, a video on-demand agent can provide video data accessing service with a bandwidth of at most  $1000KBps$ , and it can serve any number of consumers as long as the total sum of the bandwidth used by these consumers doesn't exceed  $1000KBps$ . In this case, *Service Broker* only checks whether the service can satisfy the minimum need of a request, and thus  $Require(R_i, S)$  equals the minimum requirement that  $R_i$  poses on  $S$ , for example, consumer A may request a video data accessing service with a rate at least  $300KBps$ , consumer B may request at least  $400KBps$  and consumer C at least  $500KBps$ . Thus the constraint condition in (2) will be expressed as:

- $300 \times Available(R_a, S, N) + 400 \times Available(R_b, S, N) + 500 \times Available(R_c, S, N) \leq 1000$ .

As a matter of fact, looking into equations (2), (3) and (4), we can see that only  $Available(R_i, S, N)$ s are variables, therefore, solving a request collision turns to be solving a simple linear programming problem with the variable domain to be  $\{0,1\}$ . The solutions can demonstrate which consumer can get the desired service when there is a collision. For example, a solution,  $Available(R_a, S, N)=0$ ,  $Available(R_b, S, N)=1$ ,  $Available(R_c, S, N)=1$ , means only consumer A can not get service S.

**Algorithm for Service Brokering.** A service request is handled by *Service Broker* as a transaction because we believe that the process of satisfying a request should be

```

function SatisfyRequest(rd) returns a service aggregation or failure;
  inputs: rd, request description from a consumer;
  return Commit(GetService(rd)); // two phase commit

function GetService(rd) returns a service aggregation;
  SA = {}; initial service aggregation
  service ← the service that matches rd;
  if service doesn't exist, then SA={}, return SA;
  if service is an atom service, then SA = SA ∪ {service}, return SA;
  if service is a composite service, then
  {
    traverse the tree of this composite service using preorder traversal;
    for each node i in the tree do
      rdx ← service description of node i;
      GetService(rdx);
  }

function Commit(SA) returns a solution or failure;
  inputs: SA, service aggregation containing all the desired atom services to
  construct a composite service;
  if SA = {}, return failure;
  reserve all the services in SA;
  if any collision occurs when reserving, then solve the collisions using linear
  programming model;
  if all services are reserved successfully, then return SA;
  if any service can not be reserved, then return failure;

```

Fig. 3. Algorithm for service brokering

indivisible, or we say atomic. *Service Broker* adopts a two-phase commit algorithm to guarantee the atomicity of the procedure of satisfying a request, in which *Service Broker* first collects and then reserves all the requested services, and according to the reservation responses from service providers, decides whether the request can be fulfilled. A short description of the algorithm is shown in Figure 3. For a composite service, when all its desired atomic services are available, *Service Broker* will take steps to coordinate agents providing these services to perform the composite service based on its knowledge description.

### 3 Conclusion and Future Work

When multi-agent systems are situated in smart spaces to address the coordination of various smart things, agent coordination approaches encounter new challenges. In this paper, we present Baton, a service management system to enhance the coordination mechanisms of multi-agent systems in smart spaces. Services in Baton are described by OWL-S language, which makes the processes of service discovery and composition more accurate and efficient. Solutions for request collisions are modeled as simple linear programming problems, which makes it easy to solve the collisions and in the meanwhile, keep changes of service dependencies to the minimum. The process of fulfilling a request is handled as a transaction, and a two-phase commit algorithm is utilized to assure its atomicity. Currently, Baton has been built into our Smart Classroom [8] and Smart Meeting Room [9] to manage the services of the systems so as to sustain better coordination of the smart things in smart spaces.

We are now trying to improve the dynamic service composition strategy by using the semantic information of services, and will add proper access controls of services to Baton so as to settle the security problem in smart spaces.

### References

1. NIST Smart Space Laboratory. <http://www.nist.gov/smartspace>
2. David L. Martin, Adam J. Cheyer, Douglas B. Moran: The open agent architecture: A framework for building distributed software systems. *Applied Artificial Intelligence*, 13(1-2): 91–128, January-March 1999
3. Brenton Phillips. *Metaglué: A programming language for multi-agent systems*. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, 1999
4. Xie W K, Shi Y C, Xu G Y, et al: Smart Platform - A Software Infrastructure for Smart Space (SISS). *The Fourth International Conference on Multimodal Interfaces*, Pittsburgh, USA, 2002
5. Andrea Omicini, Sascha Ossowski: Objective versus Subjective Coordination in the Engineering of Agent Systems. *The AgentLink Perspective*: pp. 179 - 202
6. Martin Fredriksson, Rune Gustavsson, Alessandro Ricci: Sustainable Coordination. *The Agent Link Perspective*: pp. 203 - 233
7. Krzysztof Gajos: *Rascal – A Resource Manager for Multi-Agent Systems in Smart Spaces*. CEEMAS01, Krakow, Poland, 2001
8. Yuanchun Shi, Weikai Xie, Guangyou Xu, et al: The Smart Classroom: Merging Technologies for Seamless Tele-Education. *IEEE Pervasive Computing*, vol. 2, no. 2, pp. 47-55, 2003

9. Xin Xiao, Enyi Chen, Yuanchun Shi: Multimedia Communication between Mobile Devices and Smart Spaces. The 13th National Multimedia Conference, Ningbo, China
10. F. Paterno, C. Mancini, S. Meniconi: Concur-TaskTrees: A Diagrammatic Notation for Specifying Task Models. Proc. Interact, Sydney, 1997
11. Berkeley DB XML Download page. <http://www.sleepycat.com/download/xml/index.shtml>
12. Robert Kochman: Decision Theoretic Resource Management for Intelligent Environments. <http://www.csail.mit.edu/research/abstracts/abstracts03/interfaces-applications/interfaces-applications.html>
13. Jade Technical Overview. <http://www.jadeworld.com/downloads/Jade6/technicaloverview>
14. Automated Office. <http://www.ai.sri.com/~oaa>
15. OWL-S 1.1 Release. <http://www.daml.org/services/owl-s/1.1>
16. XML Path Language Version 1.0. <http://www.w3.org/TR/xpath>

# An Extensible Ubiquitous Architecture for Networked Devices in Smart Living Environments

Thierry Bodhuin, Gerardo Canfora, Rosa Preziosi, and Maria Tortorella

RCOST - Research Centre On Software Technology,  
Department of Engineering, University of Sannio,  
Via Traiano, Palazzo ex-Poste – 82100, Benevento, Italy  
{bodhuin, canfora, preziosi, tortorella}@unisannio.it

**Abstract.** Continuous technological innovation is entailing that living environments be equipped with products that improve the quality of daily life. Unluckily, the adopted solutions do not always represent an adequate support and people continue to execute repetitive tasks that software infrastructures could perform automatically. This is partially due to the fact that the existent technological solutions cannot be always integrated in a coherent communication platform, as they use proprietary protocols and ad hoc implementations not easily reusable. This paper proposes an extensible ubiquitous architecture for networked virtualized devices in smart living environments. The aim is realizing ubiquitous applications and integrating networked devices through an architecture that hides their complexity and heterogeneity. Several intelligence techniques have been integrated for offering a smart environment through the use of automatic learning techniques.

## 1 Introduction

People thirst for technological products helping them to have a better quality of the everyday life. They equip with these products their professional, personal, transit, transport, and so on, living environments. Academic and industrial world feel inclined to promote technological progress and terms as *home automation*, *domotic system*, *smart home* are diffusing in the industry, while expressions as *pervasive computing*, *ubiquitous computing*, *nomadic computing*, *ambient intelligent*, *context-aware computing*, *augmentation of the real world*, indicate academic research topics. The available technologies do not always represent an adequate support as they are often unable to interact with other products made by different makers and/or based on different solutions. Their communication and integration too often requires human intervention, and people are discouraged by the complexity of the new Information and Communication Technologies (ICT) facilities and by the redundancy of the needed administrative and configuration activities. In addition, the use of proprietary communication protocols in software architectures for smart environments does not facilitate the interoperability of the networked devices and the reusability and maintainability of software packages forming part of the architecture. This forces developers to perform repetitive implementation tasks.

The work presented in this paper has been carried on within the *Demonstrator* project of the Regional Centre of Competence in Information and Communication Technology, CRdC ICT. This Centre involves many researchers and industrial partners of the Campania Region in Italy. It aims at analysing, defining and realizing hardware and software platforms for permitting the provision of networked services and the implementation of advanced technologies. In particular, the activities carried on in the unit of the University of Sannio, RCOST (Research Centre On Software Technology), aim at developing a platform in the field of home automation that is endowed with different levels of intelligence. It addresses the following aspects: *virtualization of devices*, for defining a generic functional characterization of the networked devices, making the applications independent from the characteristics of a particular device and supporting implementation tasks of software developers; *abstract description of devices*, for defining a semantic characterization of the networked devices, making applications more aware of the triggered actions in the physical world and supporting human intervention and interaction; *abstract description of services*, for providing a functional and semantic characterization of the services with reference to their relations with the other services and devices.

The proposed software architecture aims at facilitating the interoperability of networked devices, based on different technologies, and produced by different manufacturers; offering a middleware supporting different levels of intelligence as awareness, reactivity and adaptiveness; and permitting to activate services, through suitable applications respect to the typology of the client accessing it.

In the following, Section 2 presents some related work, Section 3 describes the software architecture, Section 4 discusses an example of virtualization, and the final section summarizes the main conclusions and sketches future directions of research.

## 2 Related Work

The increasing request of telecommunication solutions conducted to the development of sophisticated networked heterogeneous devices, supporting one or more of the available communication protocols (e.g., X-10, EIB, LonWorks, Ethernet-TCP/IP) and/or service and discovery-focused standards (e.g., HAVi, Jini, OSGi, UPnP). Currently, these standards are complementary, rather than competitive, even if they are sometimes partially overlapped in some provided facilities. The use of networked devices supporting different protocols and standards requires the adoption of more complex networking techniques, facilitating the interaction and interoperability of the devices and their accessibility from both local-area and wide-area networks.

In this scenario, it would be expected that different interconnected networks, supporting distinct features of smart living environments, exist. Consequently, manufacturers of different communication technologies, such as LonWorks and EIB, continuously upgrade their systems for increasing the reciprocal interoperability [4] and allowing devices from different vendors to communicate each other. However, the communication between devices is still not supported [4, 10, 11] in many cases. For example, it is possible to find living environments including EIB controlled devices, Ethernet networked devices and Bluetooth mobile devices, but it is unlikely to find living environments where other components, such as a X-10 and a EIB controlled lamp, interoperate.

In many cases, the effort addresses the integration of various physical elements, including sensors, actuators, microcontrollers, computers and connectors [5]. But, many of the proposed solution are mostly manual and ad-hoc, lack of scalability and are too close to the third parties. Likewise, each time a new component is inserted into the considered space, conflicts and uncertain behaviours may be verified in the overall system, requiring programming and testing interventions. For facing these problems, a middleware automating integration task is required for ensuring pervasive space openness and extensibility [6]. It must enable programmers to dynamically integrate devices without interacting with the physical world, and, then, decouple programming tasks from construction and integration of physical devices.

The typical approach that is applied regards the connection of sensor-actuator devices using classical network infrastructures, such as OSI, CORBA, and so on, at a low level. Unfortunately, the use of these kinds of infrastructures does not ease the integration of the devices. The approach in [12] is based on the integration of the devices at high-level, and ad-hoc networking infrastructures that dynamically integrate sensors and actuators into complex interactive systems while providing services and interfaces.

The architectural design presented in this paper has been defined for partially solving the problems of integrating devices, and for controlling and monitoring personal living environments from heterogeneous terminals. It considers requirements of *interoperability*, *portability*, *extensibility*, *reusability* and *maintainability* from the developer's point of view and *usability* and *adaptability* from the end-user's point of view. In addition, the proposed solution is based on the *OSGi (Open Service Gateway initiative)* [9], an emergent open architecture, which permits the deployment of a large array of wide-area-network services to local network services such as smart homes and automobile [5]. OSGi defines a lightweight framework for delivering and executing service-oriented applications. It presents advantages, such as: platform independence, different levels of system security, hosting of multiple services and support for multiple home-networking technologies.

### 3 Extensible and Ubiquitous Architectural Design

Figure 1 shows the proposed extensible ubiquitous architectural design. The various layers are grouped in six levels, going from **A** to **F**, and they are next presented.

#### 3.1 Levels F, E, D

Level **F** in Figure 1 depicts the heterogeneous networked devices to be accessed. They may be produced from different manufacturers and/or using different communication protocols and, service and discovery-focused standards. Level **E** includes the needed drivers, grouped in two layers: a hardware layer and a layer of network IP cards, audio cards, RS-232 ports, etc., necessary for connecting the devices of level **F**. Level **D** concerns the portability of the implemented software and includes the operating system and the Java Virtual Machine (JVM).

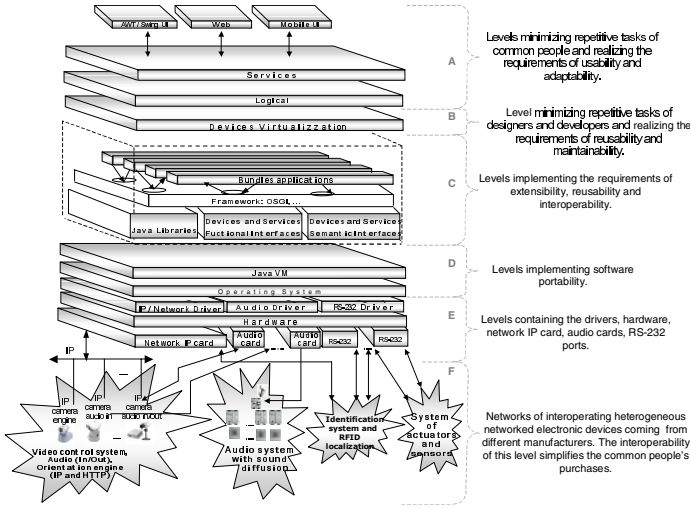


Fig. 1. Extensible and ubiquitous architectural design

### 3.2 Levels A, B, C

Levels A, B and C form the *Domus intelligent Keeper (DiK)* software infrastructure. *DiK* is composed of three main components: a *framework* component, which aims at minimizing the activities of developers and helping the extensibility and ubiquity capability of the architecture; a *service oriented applications* component which uses the framework and aims at simplifying and minimizing human intervention and interaction activities; and an *intelligence* component aiming at decreasing repetitive daily activities and facilitating the automatic evolution of the software infrastructure, on the basis of people’s continuous changing habits and modifications of the networked devices adopted in the living environments.

Level C assures the characteristic of interoperability of the proposed architectural design. It includes the *OSGi (Open Service Gateway initiative)* Framework [9] enabling the connectivity and management of the devices based on different transmission technologies. It defines a platform model where the software applications are installed and executed. These applications are Java archives, called *bundles*, which cooperate to the implementation of a service. The OSGi Framework represents a common environment hosting bundles. The bundles use: the resource of the OSGi Framework, all the standard Java libraries, virtualized devices and service interfaces. In addition, they access level E and, consequently, monitor and control the networked devices of level F. In particular, the OSGi Framework is the part that changes a JVM from a single application environment into a multiple one. The advantages are many. Actually, running multiple applications in a single JVM means less process swaps, fast inter-application communication, and significantly less memory consumption. Moreover, the OSGi Framework makes possible the interoperability among different devices, service providers, network operators, service gateway manufacturers, and home appliance manufacturers. Level C in Figure 1 manages the life cycle of the bundles and solves their interdependence, keeps a



registry of services and manages the events informing the listeners when the state of a bundle is changed, when a service is stored and when an error occurs. Besides the usage of OSGi bundles, level C includes an alternative device access solution based on the *Java Remote Method Invocation (RMI)* and the Jini technology. RMI/Jini and OSGi solutions are not the only ones to be considered for level C, as this level is a dynamic container with changeable content according to the technological progress so that it can deliver access to services over any network and protocol. Level C was developed with the intention of ensuring the satisfaction of the extensibility requirement. This aspect is strongly related to the capability of evolving the software when new technologies are introduced and needs of the end-user change. The extensibility requirement is also preserved by the usage of class libraries in the level C. In this manner, developers can take advantage from the object-oriented techniques, which facilitate a more modular designing and encourage the use of constructs related to inheritability for better organizing the source code, avoiding repetitions, gaining time and reducing development costs.

Level B, including the *Devices Virtualization layer*, is located between the bundles of the OSGi Framework and Level A of the services accessible from the user. Its objective is to provide an abstraction of the devices of level F, by generalizing their behaviour independently from their identity (or type), nature and communication protocol, and hiding the complexity of the reciprocal communications. In particular, two different devices have different identity expressed from a set of attributes like: *name, serial, version, model, manufacturer*, etc. Two devices with different nature are logically connected to two distinct physical concepts. Nevertheless, two different devices with distinct type and nature may share the same actuation mechanism. For example, a networked lamp is a device different from an alarm. The lamp is logically connected to the *electric light* concept and may change the state of the environment where it is installed by providing, or not providing, light on the basis of the switch *on/off* actuation mechanism. The alarm is logically connected to the *sound* concept and may change the state of the environment hosting it by providing or not providing noise in accordance with its *open/close* actuation mechanism. The lamp and alarm are devices of different identity, nature and semantic, but share an actuation mechanism with the same working procedure. So, it is possible to extract a functional view permitting a first classification of the devices grouping them in two families: *Sensors*, capturing information from the networked devices and/or the environments, and producing events; *Actuators*, consuming events and, triggering actions on the networked devices in the considered environments. Sensors and Actuators can be still specialized in other objects. For example, the networked rolling shutter has a mechanism of actuation different from that of the networked lamp and alarm. It cannot be defined by two values but considering a set of valid values. For instance, the rolling shutter may have five possible valid values, *absent, low, medium, high, highest*, modelling five different positions and brightness degrees. Besides the Sensors and Actuators, complex devices exist in the living environments. They are the result of the composition of more elementary devices. For instance, a camera is defined as a complex device with different elementary actuation mechanism related to different functionalities, as later described. Figure 2 exhibits a view of the device interface hierarchy. It shows that the specialization of the generic devices of type Sensor and Actuator is reasonable. For example, the networked lamp is a device of

Actuator type, which can be described by a *BinaryActuator* interface, able to assume only two valid values. While the rolling shutter is a device of Actuator type describable by a *SetValuesActuator* interface being able to assume different discrete defined values. Furthermore, a device with values inside a given continuous range can be characterized by a *RangeValuesActuator* interface. Besides those discussed, further specialization levels can be identified. In addition, Figure 2 highlights that Interface *Device* is characterized by methods adding/removing the *EventListener* objects and used from clients for registering/un-registering a listener in *Device*. Thus, clients can be notified in a push way of changes in the state of the devices for taking their decisions. Listener and event interface hierarchies are also defined. Moreover, interface *Device* is characterized by getting/ setting methods for accessing and/or manipulating the *identity* of a considered device. The identity information is maintained in the logical layer and its handling is a first step toward the modelling of devices that considers the semantic aspect.

The interface hierarchy shown in Figure 2 is not complete. It permits the realization of reusable software components. Furthermore, the Devices Virtualization layer is still valid, even when the hierarchy is extended for including new devices, independently from their complexity.

Finally, level **A** groups the layers oriented to minimize the work of the end-user. In fact, they allow *DiK* to adapt a personal living environment to the needs of common people and/or situations and to simplify the human interaction. Level **A** includes three layers named *Logical*, *Services* and *User Interface (UI)*.

*Logical* layer manages and maintains the information regarding the logical *internal* characterization of each networked electronic device and the optional logical *external* characterization. The internal characterization of a device is defined by its datasheet, while the external one is described by the *logical connections* between the considered device and the physical concepts it can affect. The physical concepts are attributes characterizing the environment that is *external* to the device. For example, a networked rolling shutter is a device internally characterized by the *raising* behaviour.

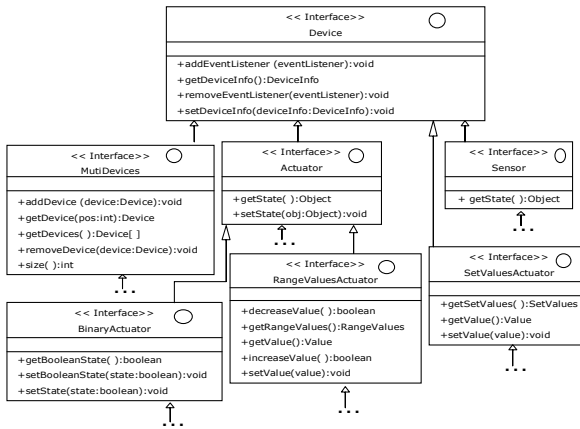


Fig. 2. A simplified view of device interface hierarchy

This behaviour allows the rolling shutter to be (un)rolled at a given grade. In this way, it allows one to change the state of brightness of a given environment. Therefore a logical connection exists between the cited device and the *solar light* physical concept, which represents its external characterization.

*Services* layer aggregates functionalities exported from single networked devices for providing services that are able to promote comfort, safety, security, initial minimization of human intervention and improved lifestyle for residential end-users. For instance, if an *illumination control service* exists in a house, it could promote comfort in terms of luminosity, while reducing associated cost for producing light in the area where the householder is located. This service may use: any localization sensor (e.g., presence sensor, RFID identification) for recognizing human presence in a given area; a luminosity sensor for knowing if a given luminosity threshold is achieved; and light actuators, like dimmer or on/off lamp, for reaching the light condition requested. Aggregating the functionality of networked curtains, rolling shutters and lamps allow the reduction of the associated cost for reaching a certain luminosity level, corresponding to the householder wished level of comfort. The control services use standard control mechanisms with loop control. However, in the context of home automation networks with slow action to effect, the control services were enhanced by using neural network for learning the relation between wishes (e.g., light condition), context (e.g., sensors, time, occupants), and possible actuations on the different actuator devices, that are located in the area where the service control takes place and are connected with the interesting physical aspect (e.g., devices connected with the illumination aspect). The use of a neural network allows the control services to achieve more rapidly their objective on slow networks and/or slow action/effect relation. In addition the Service layer includes a group of intelligent services permitting to support different levels of intelligence: context-aware, automated, reactive, adaptive. Whatever intelligence type might be, it requires the measurement and collection of data, as well as the extraction, aggregation and abstraction of information. The progress made in hardware technology allows storing very large amounts of data, while abstracting valuable information is still a very difficult task. This task is more difficult when applied to data collected when the people interact with devices and services in the living environments. A high degree of randomness in the real human life is source of high complexity.

Despite the high degree of randomness, it is possible to identify patterns in the person's life [3]. Patterns may represent regular repetitive interactions of the people with the networked devices. People have habits that are usually sampled in time and inter-connected with the other people's habits through various constraints, which are dependent on the current role and activities that people have when they use the devices and services of the actual environment. A person's life can be "sampled" on the basis of the hours, days, week days or week-ends, seasons, and so on. Human living environments can be "sampled" based on the location, room or areas, where federations of devices and persons are defined. The repetition of these patterns may have a high or low frequency according to the variability of the person's life. These facts suggest that person's life in human living environments can be automatically "photographed" and patterns captured, processed and transformed in rules for enabling control systems and autonomously acting, while remaining unobtrusive, in addressing people's needs by requesting user's feedback.

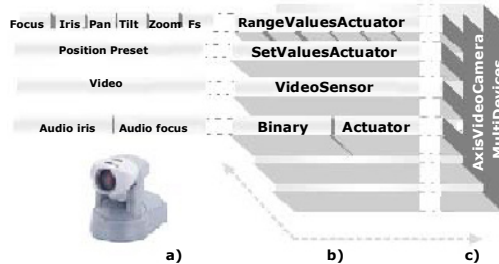
One important component of the intelligent services group is a rule engine named Jess [8] that allow the execution of rules describing relations between events and actions. The rules may be created by smart environment users, or be automatically generated by a learning system that was developed on the basis of the WEKA (Waikato Environment for Knowledge Analysis) tool [14]. This tool provides a suite of facilities for applying data mining techniques to large data sets for supporting various tasks including classification, market basket analysis (MBA or association rules), prediction. Currently, MBA algorithms are used for analyzing end-user patterns.

*User Interface* layer allows a transparent access to heterogeneous networked devices installed in living environments from interface AWT/Swing, Web and mobile.

### 4 An Example

Figure 3 depicts an example of virtualization. It refers to an Axis Video Camera with Pan/Tilt and Zoom functionalities [1]. The figure is organized in three blocks going from a) to c).

Block b) shows that the Java *AxisVideoCamera* class is implemented as a specialization of the *MultiDevices* class. In particular, it is composed of the following parts: six *RangeValuesActuator*, which are specializations of the *Actuator* class and virtualize the Pan, Tilt, Zoom, Iris, Focus and Frame/sec functionalities; one *SetValuesActuator*, which is a specialization of the *Actuator* class and virtualizes the preset position functionality; two *BinaryActuator*, which are specializations of the *Actuator* class and virtualize the Auto iris and Auto focus functionalities; one *VideoSensor*, virtualizing the video functionality as a specialization of the *Sensor* class. The specific implementation of *VideoSensor* for the Axis Video Camera includes the implementation of the Java Media framework *DataSource* [7] for the encapsulation of the MJPEG format provided by the Web server included in the Internet video camera. All the implemented classes include the functional code needed for the communication between the specific Actuators and Sensors and the physical Axis Camera, in accordance to the contract between the device implementation and their clients. Further, they exhibit suitable interfaces, exemplified by the c) block, to the client objects. The implementation of the considered Axis Camera uses the same actuation mechanisms adopted in other devices, such as the networked lamp, alarm and rolling shutter, but with a different semantic specification.



**Fig. 3.** Camera virtualization: a) mask of functionalities; b) implemented classes; c) interfaces

Therefore, a new complex device, different from the Axis camera, can be obtained simply changing the mask of the functionality shown in block **a**). This is possible thanks to the device virtualization process. In particular, the implementation of the defined classes are generic and provide a generic implementation of the methods for getting and setting the state of a device, for the (un)registering of listeners and events notification, related to the modification of the state of a device. Therefore, the device virtualization process simplifies the reuse of the generic parts of the devices and the mechanisms notifying change events to the listeners. When the implemented classes and their interfaces are introduced in the framework of the architectural design shown in Figure 1, it is possible to get and modify the video camera state trough any kind of user interfaces. For example, the Axis Video camera provides an http network protocol interface. Getting and modifying the video camera state (e.g., Rotating the video camera in PAN/TILT or Zooming), could be performed through the http interface, and connecting to the URL <http://videocamerahost/axis-cgi/com/ptz.cgi?autofocus=on>, sets the state of the *BinaryActuator* regarding the AutoFocus to ON. The Video source is acquired in a MJPEG format from an http connection to the networked video camera (e.g., <http://videocamerahost/axis-cgi/mjpg/video.cgi>). This video source is encapsulated inside a Java Media Framework DataSource for facilitating its integration with the video/audio streaming and the visualization utilities offered by the Java Media Framework. The video source is transmitted by using the Real Time Streaming protocol for permitting its visualization through unicast or multicast connection and in on-demand way. The device virtualization process also simplifies the implementation of the device remotization for letting it be accessible in a remote way by using a RMI interface. The actual protocol between the RMI client and server is defined through the Jini Extensible Remote Invocation [13] that permits the use of protocols different from the Java specific one, named JRMP.

The described implementation was tested with frame rate of more than 30 frame/sec through Real Time Streaming protocol and replicated with a D-Link DCS 2100+ Wireless Internet Audio/Video Camera, providing the video and audio without the Pan/Tilt and Zoom functionalities.

## 5 Conclusions and Future Work

This paper describes an extensible ubiquitous layered architectural design for smart living environments supporting different levels of intelligence. The technologies used for developing it, were already used with success in other projects in the ubiquitous computing context. The main difference respect to the previous usage consists of the existence of the **B** layer. It contains the Devices Virtualization layer and is oriented to decouple the **A** layers from layers below it. So, several technologies can be integrated for providing an architecture that is open to different makers and adequately supports the developers implementation tasks and decisions of the users that can feel free to buy and insert different new devices in their living environments and make them operative. Devices Virtualization layer aims at defining a framework for easily developing services, by decoupling the physical devices from the clients accessing them, and offering a middleware that permits the activation of a service, choosing a suitable user interface implementation with reference to the type of client accessing it. Further, this layer enables DiK to better survive to the changes due to the

technological progress. This aspect is very important when a software system with unstable requirements has to be developed. This is the case of the applications for living environments, where people's habits continuously change together with the physical devices to be used and integrated.

The need of a semantic characterization for networked devices was also highlighted, for addressing the dynamic discovery of devices and services, promoting comfort, safety, security, communication, and so on. This aspect is deepened in [2]. It required investigation in using ontology and specialized representation mechanisms of contextual information for ubiquitous systems. Finally, the Intelligence services were developed to achieve automatic generation of rules based on the finding of patterns in the interaction between users and devices/services in the smart living environment. Another Intelligence service regarded finding the relations for each physical aspect (e.g., light, temperature) between sensor level target and possible actuations considering constraints like cost saving. The Intelligence services use data-mining and neural networks techniques and apply them for achieving smart living environments without creating autonomous and non-manageable or understandable environment.

Future work will be considered in the field of embedded software in hardware devices with distributed infrastructure and intelligence. The aim is to support the cooperation between these devices to reach some comfort level based on the living environment occupants without needing of a semi-centralized architecture.

## References

1. Axis Communications: Axis Networked Video Camera. [http://www.axis.com/products/cam\\_213/](http://www.axis.com/products/cam_213/)
2. Bodhuin, T., Canfora, G., Preziosi, R., Tortorella, M.: Hiding complexity and heterogeneity of the physical world in smart living environments. Submitted. Available from the authors (2005)
3. Eagle, N., Pentland, A.: Reality Mining: Sensing Complex Social Systems, J. of Personal and Ubiquitous Computing. To appear (2005). <http://reality.media.mit.edu/pdfs/realitymining.pdf>
4. Fuertes, C. T.: Automation System Perception-First Step towards Perceptive Awareness Dissertation. Institute of Computer Technology, TU Wien (July 2003)
5. Gu, T., Pung, H.K., Zhang, D. Q.: Toward an OSGi-Based Infrastructure for Context Aware Applications, IEEE Pervasive Computing, Vol.3, No.4 (October-December 2004) 66-74
6. Helal, S.: Programming Pervasive Spaces, IEEE Pervasive Computing, Vol.4, No.1 (January-March 2005) 84-87.
7. JavaSoft: Java Media Framework. <http://java.sun.com/products/java-media/jmf/index.jsp>
8. Sandia National Laboratories: Java Expert System Shell. <http://herzberg.ca.sandia.gov/jess>
9. Open Service Gateway Initiative: The Open Service Gateway. <http://www.osgi.org>
10. Russ, G.: Situation-dependent behaviour in building automation. Dissertation, Institute of Computer Technology, TU Wien (2003)
11. Russ, G., Dietrich, D., Tamarit, C.: Situation Dependent Behaviour in Building Automation. Proceedings of Workshop EurAsia-ICT 2002, Advances in Information and Communication Technology, Shiraz, Iran (2002) 319-323
12. Schramm, P., Naroska, E., Resch, P., Platte, J. Linde, H. , Stromberg, G. and T. Sturm,: A Service Gateway for Networked Sensor Systems, IEEE Pervasive Computing, Vol.3, No.1 (January-March 2004) 66-74
13. Sommers, F.: Call on extensible RMI – An Introduction to JERI, JavaWorld. [http://www.javaworld.com/javaworld/jw-12-2003/jw-1219-jiniology\\_p.html](http://www.javaworld.com/javaworld/jw-12-2003/jw-1219-jiniology_p.html) (2003)
14. Waikato Environment for Knowledge Analysis Project. <http://www.cs.waikato.ac.nz/~ml/>

# A World Model for Smart Spaces

Ichiro Satoh

National Institute of Informatics,  
2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan  
ichiro@nii.ac.jp

**Abstract.** A world model for ubiquitous computing environments is presented. It can be dynamically organized like a tree based on geographical containment, such as in a user-room-floor-building hierarchy and each node in the tree can be constructed as an executable software component. It provides a unified view of the locations of not only physical entities and spaces, including users and objects, but also computing devices and services. A prototype implementation of this approach was constructed on a Java-based mobile agent system.

## 1 Introduction

Various computing and sensing devices are already present in almost every room of a modern building or house and in many of the public facilities of cities. As a result, spaces are becoming perceptual and smart. For example, location-sensing technologies, e.g., RFID, computer vision, and GPS, have been used to identify physical objects and track the positions of objects. These sensors have made it possible to detect and track the presence and location of people, computers, and practically any other object we want to monitor. There have been several attempts for narrowing gaps between the physical world and cyberspaces, but most existing approaches or infrastructures inherently depend on particular sensing systems and have inherently been designed for their initial applications.

A solution to this problem would be to provide a general world model for representing the physical world in cyberspaces. Although several researchers have explored such models, most existing models are not available for all ubiquitous computing, because these need to be maintained in centralized database systems, whereas the environments are often managed in an ad-hoc manner without any database servers. We also need often necessary to maintain computing devices and software in addition to modeling entities and spaces in the physical world. This paper focused on discussing the construction of such a model, called *M-Spaces*, as a programming interface between physical entities or places and application-specific services in cyberspaces in ubiquitous computing environments.

## 2 Background

Many researchers have explored world models for ubiquitous computing environments. Most existing models have been aimed at identifying and locating entities, e.g., people

and physical objects and computing devices in the physical world. These existing models can be classified into two types: physical-location and symbolic-location models. The former represents the position of people and objects as geometric information, e.g., NEXUS [5, 2] and Cooltown [6]. A few applications like moving-map navigation can easily be constructed on a physical-location model with GPS systems. However, most emerging applications require a more symbolic notion: place. Generically, place is the human-readable labeling of positions. The latter represent the position of entities as labels for potentially overlapping geometric volumes, e.g., names of rooms, and buildings, e.g., Sentient Computing [4], and RAUM [3]. Existing approaches assume that their models are maintained in centralized database servers, which may not always be used in ubiquitous computing environments. Therefore, our model should be managed in a decentralized manner and be dynamically organized in an ad-hoc and peer-to-peer manner. Virtual Counterpart [7] supports RFID-based tracking systems and provides objects attached to RFID-tags with Jini-based services. Since it enables objects attached to RFID-tags to have their counterparts, it is similar to our model. However, it only supports physical entities except for computing devices and places. Our model should not distinguish between physical entities, places, and software-based services so that it can provide a unified view of ubiquitous computing environments, where not only physical entities are mobile but also computing devices and spaces.

The framework presented in this paper was inspired by our previous work, called SpatialAgents [10], which is an infrastructure that enables services to be dynamically deployed at computing devices according to the positions of people, objects, and places that are attached to RFID tags. The previous framework lacked any general-purpose world model and specified the positions of physical entities according to just the coverage areas of the RFID readers so that it could not represent any containment relationship of physical spaces, e.g., rooms and buildings. Moreover, we presented another location model, called *M-Space* [11] and the previous model aimed at integrating between software-based services running on introducing computing devices and service-provider computing devices whereas the model presented in the paper aims at modeling containment relationship between physical and logical entities, including computing devices and software for defining services.

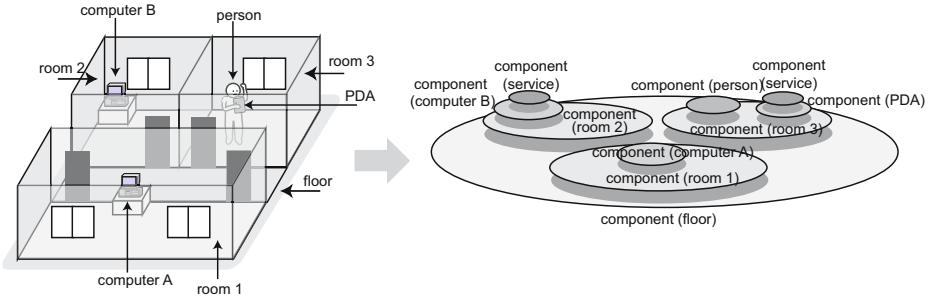
### 3 World Model

This section describes the world model presented in this paper. The model manages the locations of physical entities and spaces through symbolic names.

**Hierarchical World Model.** Our model consists of elements, called components, which are just computing devices or software, or which are implemented as virtual counterpart objects of physical entities or places. The model represents facts about entities or places in terms of the semantic or spatial containment relationships between components associated with these entities or places.

- **Virtual counterpart:** Each component is a virtual counterpart of a physical entity or place, including the coverage area of the sensor, computing device, or service-provider software.





**Fig. 1.** Rooms on floor in physical world and counterpart components in location model

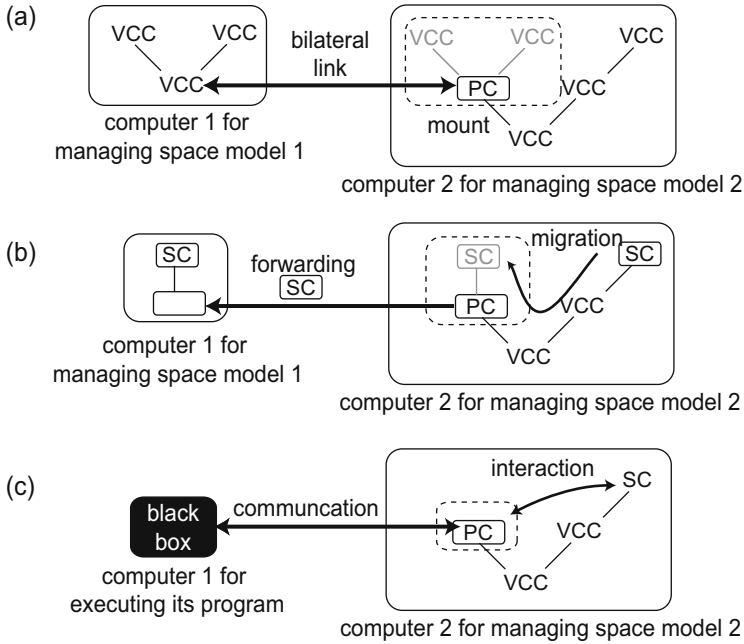
- **Component structure:** Each component can be contained within at most one component according to containment relationships in the physical world and cyberspace.
- **Inter-component movement:** Each component can move between components as a whole with all its inner components.

When a component contains other components, we call the former component is called a *parent* and the latter *children*, like the MobileSpaces model [8]. When physical entities, spaces, and computing devices move from location to location in the physical world, the model detects their movements through location-sensing systems and changes the containment relationships of components corresponding to moving entities, their source and destination. Each component is a virtual counterpart of its target in the world model and maintains the target's attributes. Fig. 1 shows the correlation between spaces and entities in the physical world and their counterpart components. The model also offers at least two basic events, entering and leaving, which enable application-specific services to react to actions in the physical world. Since each component in the model is treated as an autonomous programmable entity, it can some defines behaviors with some intelligence.

**Components.** The model is unique to existing world models because it not only maintains the location of physical entities, such as people and objects, but also the locations of computing devices and services in a unified manner. As we can see from Fig. 2, components can be classified into three types.

- **Virtual Counterpart Component (VCC)** is a digital representation of a physical entity, such as a person or object, except for a computing device, or a physical place, such as a building or room,
- **Proxy Component (PC)** is a proxy component that bridges the world model and computing device, and maintains a subtree of the model or executes services located in a VCC.
- **Service Component (SC)** is software that defines application-specific services dependent on physical entities or places.

For example, a car carries two people and moves from location to location with its occupants. The car is mapped into a VCC on the model and this contains two VCCs that



**Fig. 2.** Three types of proxy components

correspond to the two people. The movement of the car is mapped into the VCC migration corresponding to the car from the VCC corresponding to the source to the VCC corresponding to the destination. Also, when a person has a computer for executing services, his or her VCC has a PC, which represents the computer and runs SCs to define the services.

Furthermore, the model also classifies PCs into three subtypes, PCM (PC for Model manager), PCS (PC for Service provider), and PCL (PC for Legacy device), according to the functions of the devices. Our model can be maintained by not only the server but also multiple computing devices in ubiquitous computing environments.

- The first component, i.e., PCM, is a proxy of a computing device maintaining a subtree of the components in the world model (Fig. 2(a)). It attaches the subtree of its target device to a tree maintained by another computing device. Some computing devices can provide runtime systems to execute services defined as SCs.
- The second component, i.e., PCS, is a proxy of the computing device that can execute SCs (Fig. 2(b)). If such a device is in a space, its proxy is contained by the VCC corresponding to the space. When a PCS receives SCs, it forwards these to the device that it refers to.
- The third component, called PCL (PC for Legacy device), is a proxy of the computing device that cannot execute SCs (Fig. 2(c)). If such a device is in a space, its proxy is contained by the VCC corresponding to the space and it communicates with the device through the device’s favorite protocols.

For example, a television, which does not have any computing capabilities, can have an SC in the VCC corresponding to the physical space that it is contained in and can be controlled in, and the SC can send infrared signals to it. A computing device can have different PCs whereby it can provide the capabilities to them.

## 4 Implementation

To evaluate the model described in Section 4, we implemented a prototype system that builds on this model. The model itself is independent of programming languages but the current implementation uses Java (J2SE or later versions) as an implementation language for components.

### Component

*Virtual Counterpart Component:* Each VCC is defined from an abstract class, which has some built-in methods that are used to control its mobility and life-cycle. It can explicitly define its own identifier and attributes.

```
class VirtualCounterComponent extends Component {
    void setIdentity(String name) { ... }
    void setAttribute(String attribute, String value){ ... }
    String getAttribute(String attribute) {...}
    ComponentInfo getParentComponent() { ... }
    ComponentInfo[] getChildren() { ... }
    ServiceInfo[] getParentServices(String name) { ... }
    ServiceInfo[] getAncestorServices(String name) { ... }
    Object execService(ServiceInfo si,
        Message m) throws NoSuchServiceException { ... }
    ....
}
```

*Proxy Component:* PCs can be classified into three classes, i.e., PCM, PCS, and PCL. Each PCM attaches a subtree maintained by its target computing device to a tree maintained by another computing device. It forwards its visiting components or control messages to its target device from the device that it is located at, and vice versa, by using the component migration mechanism. Each PCS is a representation of the computing device that can execute SCs. It automatically forwards its visiting SCs to its target device by using the component migration mechanism. Each PCL supports a legacy computing device that cannot execute SCs due to limitations with its computational resources. It is located at a VC corresponding to the space that contains its target device. It establishes communication with its target device through its favorite approach, e.g., serial communications and infrared signals. For example, a television, which does not have any computing capabilities, can have an SC in the VC corresponding to the physical space that it is contained in and can be controlled in, and the SC can send infrared signals to it.

*Service Component (SC):* Many computing devices in ubiquitous computing environments only have a small amount of memory and slower processors. They cannot

always support all services. Here, we introduce an approach to dynamically installing upgraded software that is immediately required in computing devices that may be running. SCs are mobile software that can travel from computing device to computing device achieved by using mobile agent technology. The current implementation assumes SCs to be Java programs. It can be dynamically deployed at computing devices. Each SC consists of service methods and is defined as a subclass of abstract class `ServiceComponent`. Most serializable JavaBeans can be used as SCs.

```
class ServiceComponent extends Component {
    void setName(String name)
    Host getCurrentHost() { ... }
    void setComponentProfile(ComponentProfile cpf) { ... }
    ....
}
```

When an SC migrates to another computer, not only the program code but also its state are transferred to the destination. For example, if an SC is included in a VC corresponding to a user, when the user moves to another location, it is migrated with the VC to a VC corresponding to the location. The model allows each SC to specify the minimal (and preferable) capabilities of PCs that it may visit, e.g., vendor and model class of the device (i.e., PC, PDA, or phone), its screen size, number of colors, CPU, memory, input devices, and secondary storage, in CC/PP (composite capability/preference profiles) form [12]. Each SC can register such capabilities by invoking the `setComponentProfile()` method.

### Component Management System

Our model can manage the computing devices that maintain it. This is because a PCM is a proxy for a subtree that its target computing device maintains and is located in the subtree that another computing device maintains. As a result, it can attach the former subtree to the latter. When it receives other components and control messages, it automatically forwards the visiting components or messages to the device that it refers to (and vice versa) by using a component migration mechanism, like PCs. Therefore, even when the model consists of subtrees that multiple computing devices maintain, it can be treated as a single tree. Note that a computing device can maintain more than one subtree. Since the model does not distinguish between computing devices that maintain subtrees and computing devices that can execute services, the former can be the latter.

Component migration in a component hierarchy is done merely as a transformation of the tree structure of the hierarchy. When a component is moved to another component, a subtree, whose root corresponds to the component and branches correspond to its descendent component is moved to a subtree representing the destination. When a component is transferred over a network, the runtime system stores the state and the code of the component, including the components embedded within it, into a bit-stream formed in Java's JAR file format that can support digital signatures for authentication. The system has a built-in mechanism for transmitting the bit-stream over the network through an extension of the HTTP protocol. The current system basically uses the Java object serialization package for marshaling components. The package does not support

the stack frames of threads being captured. Instead, when a component is serialized, the system propagates certain events within its embedded components to instruct the agent to stop its active threads.

People should only be able to access location-bound services, e.g., printers and lights, that are installed in a space, when they enter it carrying their own terminals or using public terminals located in the space. Therefore, this model introduces a component as a service provider for its inner components. That is, each VC can access its neighboring components, e.g., SCs and PCs located in the parent (or an ancestor) of the VC. For example, when a person is in the room of a building, the VC corresponding to the person can access SCs (or SCs on PCs) in the VC corresponding to the room or the VC corresponding to the building. In contrast, it has no direct access over other components, which do not contain it, for reasons of security. Furthermore, like Unix's file-directory, the model enables each VC to specify its owner and group. For example, a component can explicitly permit descendent components that belong to a specified group or are owned by its user to access its services, e.g., PCs or SCs.

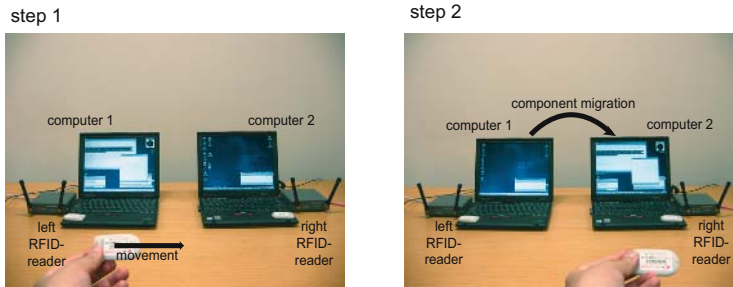
**Location-Sensor Management.** The model offers an automatic configuration mechanism to deploy components by using location-sensing systems. To bridge PCMs and location-sensors, the model introduces location-management systems, called LCMs, outside the PCMs. Each LCM manages location sensors and maintains a database where it stores bindings between references of physical entities in sensors, e.g., the identifiers of RFID tags attached to the entities and the identifiers of VCCs corresponding to the entities. Each LCM is responsible for discovering VCCs bound to entities or PCs bound to computing devices within the coverage areas of the sensors that it manages. When an entity (or device) attached to an RFID-tag and an LCM detect the presence of the entity (or device) within the coverage area of an RFID reader managed by the LCM, the LCM searches its database for VCCs (or PCs) bound to the entity (or device) and informs computing devices that maintain the VCCs (or PCs) about the VCC corresponding to the reader. Then the VCCs (or PCs) migrate to the reader's VCC. If the LCM's database does not have any information about the the entity (or device), it multicasts query messages to other LCMs. If other LCMs have any information about the entity, the LCM creates a default VCC as a new entity. When the tag is attached to an unknown device that can maintain a subtree or execute SCs, the LCM instructs the VCC that contains the device to create a default PCM or PCS for the device.

## 5 Applications

This section briefly discusses how the model represents and implements typical applications and what advantages the model has.

### 5.1 Follow-Me Applications

Follow-me services are a typical application in ubiquitous computing environments. For example, Cambridge University's Sentient Computing project [4] enabled applications to provide a location-aware platform using infrared-based or ultrasonic-based locating



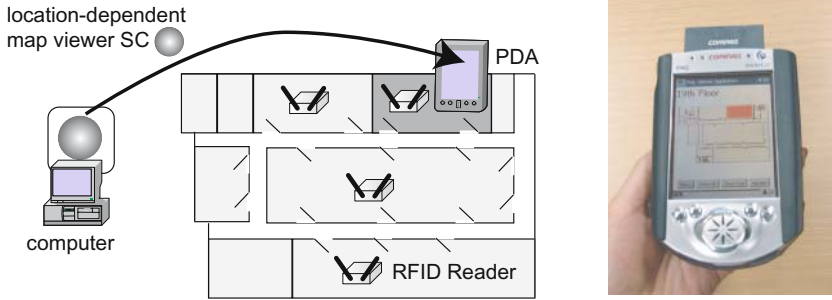
**Fig. 3.** Follow-me desktop applications between two computers

systems in a building.<sup>1</sup> While a user is moving around, the platform can track his or her movement so that the graphical user interfaces of the user's applications follow the user. The model presented in this paper, on the other hand, enables moving users to be naturally represented independently of location-sensing systems. Unlike previous studies on the applications, it can also migrate such applications themselves to computers near the moving users. That is, the model provides each user with more than one VCC and can migrate this VCC to a VCC corresponding to the destination. For example, we developed a mobile window manager, which is a mobile agent and could carry its desktop applications as a whole to another computer and control the size, position, and overlap in the windows of the applications. Using the model presented in this paper, the window manager could be easily and naturally implemented as a VCC bound to the user and desktop applications as SCs. They could be automatically moved to a VCC corresponding to the computer that was in the current location of the user by an LCM and could then continue processing at the computer, as outlined in Fig. 3.

## 5.2 Location-Based Navigation Systems

The next example is a user navigation system application running on portable computing devices, e.g., PDAs, tablet-PCs, and notebook PCs. The initial result on the system was presented in a previous paper [10]. There has been a lot of research on commercial systems for similar navigation, e.g., CyberGuide [1] and NEXUS [5]. Most of those have assumed that portable computing devices are equipped with GPSs and are used outdoors. Our system is aimed at use in a building. As a PDA enters rooms, it displays a map on its current position. We have assumed that each room in a building has a coverage of more than one RFID reader managed by an LSM, the room is bound to a VC that has a service module for location-based navigation, and each PDA can execute service modules and is attached to an RFID tag. When a PDA enters a room, the RFID reader for the room detects the presence of the tag and the LSM tries to discover the component bound to the PDA through the procedure presented in the previous section. After it has information about the component, i.e., a PCS bound to a PDA, it informs to the

<sup>1</sup> The project does not report their world model but their systems seem to model the position of people and things through lower-level results from underlying location-sensing systems.



**Fig. 4.** RFID-based location-aware map-viewer service and location-aware map-viewer service running on PDA

VC corresponding to the room about the capabilities of the visiting PDA . Next, the VC deploys a copy of its service module at the PCS and then the PCS forwards the module to the PDA to which it refers to display a map of the room. When the PDA leaves from the room, the model issues events to the PCS and VC and instructs the PCS to returns to the VC. Fig. 4 (right) outlines the architecture for the system. Fig. 4 (left) shows a service module running on a visiting PDA displaying a map on the PDA’s screen.

### 5.3 Software Testing for Location-Based Services

To test software for location-based services running on a portable device, the developer often has to carry the device to locations that a user’s device may move to and test whether software can connect to appropriate services provided in the locations. We developed a novel approach to test location-aware software running on portable computing devices [9]. The approach involves a mobile emulator for portable computing devices that can travel between computers, and emulates the physical mobility and re-connection of a device to sub-networks by the logical mobility of the emulator between sub-networks. In this model, such an emulator can be naturally implemented as a PC, which provides application-level software, with the internal execution environments of its target portable computing devices and target software as SCs. The emulator carries the software from a VCC that is running on a computer on the source-side sub-network to another VCC that is running on another computer on the destination-side sub-network. After migrating to the destination VCC, it enables its inner SCs to access network resources provided within the destination-side sub-network. Furthermore, SCs, which were tested successfully in the emulator, can run on target computing devices without modifying or recompiling the SCs. This is because this model provides a unified view of computing devices and software and enables SCs to be executed in both VCCs and PCs.

## 6 Conclusion

We presented a world model for context-aware services, e.g., location-aware and personalized information services, in ubiquitous computing environments. Like existing

related models, it can be dynamically organized like a tree based on geographical containment, such as a user-room-floor-building hierarchy and each node in the tree can be constructed as an executable software component. It also has several advantages in that it can be used to model not only stationary but also moving spaces, e.g., cars. It enables context-aware services to be managed without databases and can be managed by multiple computers. It can provide a unified view of the locations of not only physical entities and spaces, including users and objects, but also computing devices and services. We also designed and implemented a prototype system based on the model and demonstrated its effectiveness in several practical applications.

## References

1. G.D. Abowd, C. G. Atkeson, J. Hong, S. Long, R. Kooper, and M. Pinkerton, *Cyberguide: A Mobile Context-Aware Tour Guide*, ACM Wireless Networks Vol. 3, pp.421–433. 1997.
2. M. Bauer, C. Becker, and K. Rothermel *Location Models from the Perspective of Context-Aware Applications and Mobile Ad Hoc Networks*, *Personal and Ubiquitous Computing*, vol. 6, Issue 5-6, pp. 322-328, Springer, 2002.
3. M. Beigl, T. Zimmer, C. Decker, *A Location Model for Communicating and Processing of Context*, *Personal and Ubiquitous Computing*, vol. 6 Issue 5-6, pp. 341-357, Springer, 2002
4. A. Harter, A. Hopper, P. Steggeles, A. Ward, and P. Webster, *The Anatomy of a Context-Aware Application*, *Proceedings of Conference on Mobile Computing and Networking (MOBICOM'99)*, pp. 59-68, ACM Press, 1999.
5. F. Hohl, U. Kubach, A. Leonhardi, K. Rothermel, and M. Schwehm, *Next Century Challenges: Nexus - An Open Global Infrastructure for Spatial-Aware Applications*, *Proceedings of Conference on Mobile Computing and Networking (MOBICOM'99)*, pp. 249-255, ACM Press, 1999).
6. T. Kindberg, et al, *People, Places, Things: Web Presence for the Real World*, Technical Report HPL-2000-16, Internet and Mobile Systems Laboratory, HP Laboratories, 2000.
7. K. Romer, T. Schoch, F. Mattern, and T. Dubendorfer, *Smart Identification Frameworks for Ubiquitous Computing Applications*, *IEEE International Conference on Pervasive Computing and Communications (PerCom'03)*, pp.253-262, IEEE Computer Society, March 2003.
8. I. Satoh, *MobileSpaces: A Framework for Building Adaptive Distributed Applications Using a Hierarchical Mobile Agent System*, *Proceedings of IEEE International Conference on Distributed Computing Systems (ICDCS'2000)*, pp.161-168, April 2000.
9. I. Satoh, *A Testing Framework for Mobile Computing Software*, *IEEE Transactions on Software Engineering*, vol. 29, no. 12, pp.1112-1121, December 2003.
10. I. Satoh, *Linking Physical Worlds to Logical Worlds with Mobile Agents*, *Proceedings of International Conference on Mobile Data Management (MDM'2004)*, IEEE Computer Society, January 2004.
11. I. Satoh, *A Location Model for Pervasive Computing Environments*, *Proceedings of IEEE 3rd International Conference on Pervasive Computing and Communications (PerCom'05)*, pp.215-224, IEEE Computer Society, March 2005.
12. World Wide Web Consortium (W3C), *Composite Capability/Preference Profiles (CC/PP)*, <http://www.w3.org/TR/NOTE-CCPP>, 1999.



# Dealing with Emotional Factors in Agent Based Ubiquitous Group Decision

Goreti Marreiros<sup>1</sup>, Carlos Ramos<sup>1</sup>, and José Neves<sup>2</sup>

<sup>1</sup> GECAD, Knowledge Engineering and Decision Support Group,  
Institute of Engineering, Polytechnic of Porto, Porto, Portugal  
{goreti, csr}@dei.isep.ipp.pt

<sup>2</sup> University of Minho, Braga, Portugal  
jneves@di.uminho.pt

**Abstract.** With the increasing globalization of economy and consequent increasing in the inter and intra organizational competitiveness, the role of groups in organizations and businesses achieve greater significance. The work, as well as the responsibility involved to reach a decision, is distributed among group members, which may be distributed geographically and may cooperate in an asynchronous way. This paper shortly presents the *WebMeeting* prototype, which is a group decision support system that supports ubiquitous group decision meetings. It is also discussed the influence of emotional factors in group decision making and it is proposed a multi-agent model to simulate ubiquitous group decision making processes, where argumentation and emotional capabilities are considered.

## 1 Introduction

The problem of group decision-making has gained great relevance in the scope of Decision Support Systems, which were initially designed as individual tools. Quickly those tools have demonstrated to be limited, in the sense that in today's organizations several persons, entities or agents are involved in most of the decision processes. In that way decision problems are considered from different points of view, with different opinions about the importance of the decision criteria (for instance, in the purchase of a car we will be able to consider criteria like price, technical characteristics, design or manufacturer).

The present business environment is characterized by the use of groups, which work in distributed environments and have to deal with uncertainty, ambiguous problem definitions, and rapidly changing information.

In order to support group work, numerous commercial and non commercial Group Decision Support Systems (GDSS) were developed in the last years (GroupSystems software; *WebMeeting* [16]; HERMES [10]; VisionQuest software). Despite the quality of these systems, they present some limitations. In our recent work we are proposing some new ideas to deal with GDSS [14], namely: the use of Multi-Agent Systems to model group participants; and the inclusion of argumentation and emotional aspects in the group decision making process.

The work described in this paper is included in ArgEmotionAgents project (POSI / EIA / 56259 / 2004 - Argumentative Agents with Emotional Behaviour Modelling for Participants' Support in Group Decision-Making Meetings), which is a project supported by FCT (Science & Technology Foundation – Portugal) envisaging the use of Multi-Agent Systems approach for simulating Group Decision-Making processes, where Argumentation and Emotion components are specially important.

This paper is organized as follows. Section 2 provides a general approach to group decision making, in particular to ubiquitous group decision making. Yet in this section it is presented the *WebMeeting* prototype (which is a group decision support system that supports ubiquitous group decision meetings) and is discussed the role of emotion in group decision meetings. In section 3 it is presented the OCC model of emotion proposed by Ortony, Clore and Collins [20]. A model to support agent based ubiquitous group decision making is proposed in section 4, this model has several components, we will focus in the emotional component. Finally section 5 presents conclusions and gives some perspectives and ideas for future work.

## 2 Group Decision

The term Group Decision Support System (GDSS) [7][12] emerged effectively in the beginning of the eighty-decade. According to Huber [8] a GDSS consists of a set of software, hardware, languages components and procedures that support a group of people engaged in a decision related meeting. A more recent definition from Nunamaker and colleagues [18] says that GDSSs are interactive computer-based environment which support concerted and coordinated team effort towards completion of joint tasks.

Generically we may say that GDSS aims to reduce the loss associated to group work (e.g. time consuming, high costs, improper use of group dynamics, etc.) and to maintain or improve the gains (e.g. groups are better to understand problems and in flaw detection, participants' different knowledge and processing skills allow results that could not be achieved individually). The use of GDSS allows groups to integrate the knowledge of all members into better decision making.

Jonathan Grudin [6] classifies the digital technology to support the group interaction in three phases: pre-ubiquitous, the proto-ubiquitous and the ubiquitous. In the pre-ubiquitous phase, that begin in the 70's, were supported face-to-face meetings. In the proto-ubiquitous phase distributed meetings were supported, this phase begun approximately at 1990. The ubiquitous phase is now getting under way and support meetings distributed in time and space. This classification is similar to DeSancits and Gallupe [3] classification to GDSS, although in this last one it is considered another type of support, the Local Decision Network that is a type of support where group members meet at the same place but at different time.

Our interest is in ubiquitous group support.

### 2.1 Ubiquitous Group Decision Making

There are many areas where ubiquitous group decision making makes sense. One of the most cited areas in literature is Healthcare since patient's treatment involves various specialists, like doctors, nurses, laboratory assistants, radiologist, etc. These

specialists could be distributed across departments, hospitals or even in different countries. The HERMES system, a web-based GDSS was tested inside this context [10]. Many of the decisions we take every day will take a new dimension if we consider that they will be resolved by a group of individuals, for instance: choice of a place to make vacations, buy a car, hire an employee and choice of a place to build a new airport. If the group members are dispersed in time and space, the need of coordination, informal and formal communication, and information share support will increase significantly. There are already some examples of GDSS that support ubiquitous decision (GroupSystems software; *WebMeeting* [16]; HERMES [10]; Vision-Quest software).

## 2.2 Emotion in Group Decision

Common sense usually tell us that a great deal of emotion can harm decision making process but, on the other hand, Rosalind Picard for instance, claims that too little emotion can impair decision making as well [21]. It seems that, in decision making processes, emotion is needed in a balanced way.

In psychological literature several examples could be found on how emotions and moods affects the individual decision making process:

- Individuals are more predisposed to recall past memories that are congruent with their present feelings.
- Positive mood tend to promote risk aversion behaviour, while negative mood promote a risk taking behaviour.
- Positive moods tend to be associated with heuristics information strategy processing, while negative moods are more related to systematic processing.

Emotion will influence the individual decisions of the group members, but during a group decision making, group members may be also influenced by the displayed emotions of other members.

The process of emotional contagion could be analysed based on the emotions that a group member is feeling or based on the group members mood [17].

A more detailed review of the influence of emotion in group decision making can be found in [13].

## 2.3 *WebMeeting* Prototype

*WebMeeting* is a GDSS that supports distributed and asynchronous meetings through the Internet (ubiquitous meetings) [16]. The *WebMeeting* system is focused on multi-criteria problems, where there are several alternatives that are evaluated by various decision criteria. Moreover the system is intended to provide support for the activities associated with the whole meeting life cycle, from the pre-meeting phase to the post-meeting phase.

The system aims at supporting the activities of two distinct types of users: ordinary group “members” and the “facilitator”. The users of *WebMeeting* can access the system from anywhere through a PC and an Internet connection.

The *WebMeeting* system is composed by the following modules: Setup, Management, Argumentation, Multi-criteria, Voting and Database. The Setup module will be

operated by a *facilitator* during the pre-meeting phase. The Multi-Criteria module is used: by the *facilitator* during the pre-meeting phase to configure the multi-criteria decision problem; and by the participants during the meeting in order to establish individual preferences. The argumentation module is based on the IBIS (Issue Based Information System) argumentation model [18] and implements an argumentation forum where group members could argue in favor or against alternatives. The Voting module is responsible for the emission of “vote bulletins”, and for the publication of results (intermediate and final). In figure 1 it is possible to see a screen of an argumentation forum of a very simple group decision (acquisition of a laptop).



Fig. 1. Argumentation forum

An interesting and somehow natural expansion of the *WebMeeting* system might involve the addition of a simulation system where it should be possible to simulate the participants of an ubiquitous group decision meeting through emotional autonomous agents. Bellow it will be described some of the approaches that can be found in literature, that use agents and in particular multi-agent systems in group decision support systems. Section 4 will present our model of an agent based support to ubiquitous decision that handles emotional aspects.

## 2.4 Agents in Group Decision Support Systems

In literature there are already descriptions of agent based GDSS, some of them will be described afterwards.

Ito and Shintani [9] propose an architecture for an agent based GDSS where, it is associated an agent to each member (human) of the decision meeting. The key idea of this system is the persuasion mechanism between agents. The persuasion in this system is already done in pairs, for instance, agent A tries to convince agent B about the choice of alternative X, if agent A succeed then they will form a group and together will start a new persuasion cycle and try to convince another agent about the choice of alternative X.

Kudenko and colleagues [11] propose a system named MIAU whose aim is to support a group of users in the decision of acquiring a good from an electronic catalogue. The catalogue items are characterized by a set of criteria (if the item of the catalogue is a car the criteria could be: price, technical characteristics, design or manufacturer, capacity of charge). MIAU intends to obtain a compromise solution that can be acceptable for all group members and for that it acquires the preference models of each user through interface agents. After this phase a mediator agent combine all the agents and try to identify negotiable aspects and to suggest what seems to be a compromise solution. The users can accept or reject the proposed solution, and that may imply updates in the individual preference models. This process is repeated until a consensual solution is found.

Hermes [10] is a web-based GDSS that supports argumentative discourses between group members. The role of agents in this system is, for instance, to provide mechanisms to validate arguments consistency as well as to weight them. Agents in Hermes are also responsible for processes related with information search, for instance recovering information from previous discussions.

### 3 OCC Model

As we have seen before, the emotional state of an individual affects its decisions and influence the emotional state of others member of the group, through the process of emotional contagion that will be discussed in section 4. As we intend to simulate group decision making through autonomous agents, it is important that those agents have some emotional characteristics, in order to approximate the simulation to the reality.

The OCC model [20] proposes that emotions are the results of three types of subjective appraisals:

1. The appraisal of the pleasantness of events with respect to the agent's goals.
2. The appraisal of the approval of the actions of the agent or another agent with respect to a set of standards for behaviour.
3. The appraisal of the liking of objects with respect to the attitudes of the agent.

Generically in the OCC model emotions are seen as valenced reactions to three different type of stimulus [20]: objects; consequence of events and action of agents. These are the three major branches of emotion types. In the branch objects we have the emotions love and hate. In the branch consequences of events we have the emotions: happy-for, gloating, pity, resentment, satisfaction, hope, fear, fears-confirmed, relief, disappointment, joy and distress. In the branch actions of agents we have the emotions: pride, shame, admiration and reproach. The model considers yet 4 compound emotions, because they are consequence of events and agents actions, which are: gratification, remorse, gratitude and anger.

The original OCC model, described above, with his 22 different types of emotions is probably, for our propose, to much fine grained. A simplified version of this theory was presented in 2003 by Ortony [19], where he considered only two different categories of emotional reactions: positive and negative. As in the original model, emotions are the results of three types of subjective appraisals (goal-based, standard-based and taste-based). In table 1 it is possible to visualize the OCC model reviewed in 2003, after the collapse of some of the original categories.

**Table 1.** Five specializations of generalized good and bad feelings (collapsed from [19])

	Positive Reactions	Negative Reactions
Undifferentiated	...because something good happened ( <b>joy</b> )	...because something bad happened ( <b>distress</b> )
Goal-based	...about the possibility of something good happening ( <b>hope</b> )	...about the possibility of something bad happening ( <b>fear</b> )
	... because a feared bad thing didn't happen ( <b>relief</b> )	... because a hoped-for good thing didn't happen ( <b>disappointment</b> )
Standard-based	... about a self-initiated praiseworthy act ( <b>pride</b> )	... about a self-initiated blameworthy act ( <b>remorse</b> )
	... about an other-initiated praiseworthy act ( <b>gratitude</b> )	...about an other-initiated blameworthy act ( <b>anger</b> )
Taste-based	... because one finds someone/thing appealing or attractive ( <b>like</b> )	... because one finds someone/thing unappealing or unattractive ( <b>dislike</b> )

The OCC model was several times used to model the implementation of emotional agents, and afterwards we will refer to some of the implementations that use it.

Bates [2] developed the OZ project in which real-time, interactive, self-animating agents were situated in simulated micro-worlds. These agents, who were based on the principles of traditional character animation, were equipped with emotions to make them believable. The module that implements emotions in the OZ project is the EM module that is based in a simplified version of the OCC model (only some emotions of the model were implemented).

Elliot [4] developed the Affective Reasoner, a multi-agent simulation model based on the OCC emotions model, where agents have the capacity to produce twenty four emotion types and express more than 1200 facial expressions. Each agent has a representation of itself and a representation of the concerns of other agents which allow them to explain the emotional episodes of others. During the simulation, agents judge events according to their attractiveness and status (unconfirmed, confirmed, and disconfirmed).

Adamatti and Bazzan in [1] describe Afrodite, a framework to simulate agents with emotions that is based on the OCC model. With this simulation framework it is possible to configure different scenarios.

El-Nasr [5] proposes the FLAME model that is a computational implementation of emotions that uses fuzzy logic and is based in a combination of the OCC model and the Roseman emotion model [22].

Despite several implementations of the OCC model, it is not exempt of critics, probably the more cited are: the fact that OCC model does not retain memory of past emotions (interactions) and the impossibility to model an emotion mixture.

## 4 The Proposed Model

As we referred in the beginning of this paper our aim is to present a multi-agent model to simulate ubiquitous group decision making considering emotional factors. In our opinion the use of Multi-Agent Systems seems to be quite suitable to simulate the behaviour of groups of people working together and, in particular, to ubiquitous group decision making modelling, because it allows [15]:

- Individual modelling – each participant of the group decision making can be represented by an agent that will interact with other agents. Agents can be modelled with social and emotional characteristics in order to become more realistic.
- Flexibility – with this approach it is easy to incorporate or remove entities.
- Data distribution – frequently, in group decision making, participants are geographically distributed.

In our previous work we identified the main agents involved in a simulation of a group decision meeting [14] and they are: Participant Agents; Facilitator Agent; Register Agent; Voting Agent and Information Agent.

In the remain text of this section we will first present the architecture of participants agents, because they represent the main role in group decision making and then we will detail one of the components of this architecture, the Emotional module.

#### 4.1 Participant Agent Architecture

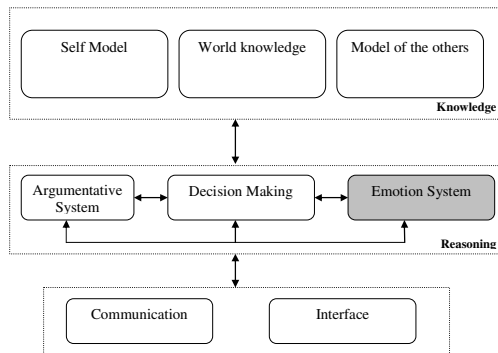
In figure 2 it is represented the architecture of participant agents. This architecture contains three main layers: the knowledge layer, the reasoning layer and the communication layer.

In the knowledge layer the agent has information about the environment where it is situated, about the profile of the other participants agents that compose the simulation group, and regarding its own preferences and goals (its own profile). The information in the knowledge layer is dotted of uncertainty and will be accurate along the time through interactions done by the agent.

The communication layer will be responsible for the communication with other agents and by the interface with the user of the group decision making simulator.

The reasoning layer contains three major modules:

- the argumentative system – that will be responsible by the arguments generation;
- the decision making module – that will choose the preferred alternative;
- the emotional system – that will generate emotions and moods, affecting the choice of the arguments to send to the others participants, the evaluation of the received arguments and the final decision.



**Fig. 2.** Participant Agent Architecture

## 4.2 Emotional Module

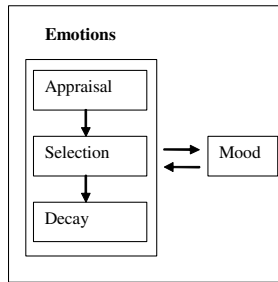
The emotions that will be simulated in our system are those identified in the reviewed version of the OCC model: joy, hope, relief, pride, gratitude, like, distress, fear, disappointment remorse, anger and dislike.

An emotion in our system is characterized by the proprieties identified in table 2.

**Table 2.** Emotion property

<b>Type</b>	Emotion type
<b>Valence</b>	Positive or negative
<b>Agent_Id</b>	Identification of the agent
<b>Time</b>	Moment in time when emotion was initiated
<b>Origin_Id</b>	Identification of the agent or event that origin the emotion
<b>Intensity</b>	Emotion intensity

In figure 3 it is possible to visualize the main components of the emotional system.



**Fig. 3.** Emotional Module

The emotional module is composed by three main components: the **appraisal** – based on OCC model, where the intensities of potential emotions are calculated; the **selection** – each emotion has a threshold activation, that can be influenced by the agent mood, this component selects the dominant emotion; and **decay** – emotions have a short duration, but they do not go away instantaneously, they have a period of decay.

The agent **mood** is calculated based on the emotions agents felt in the past and in the moods of the remaining participants. In our approach only the process of mood contagion is being considered, we do handle the process of emotions contagion. We consider only three stages for mood: positive, negative and neutral.

In group decision simulation the participant agents will exchange arguments in order to achieve a consensual solution, the selection of arguments to be sent and the evaluation of received arguments will take into account the agent internal emotional state, the moods of other agents, as well as other characteristics that compose the agents profile: debts of gratitude, agents in which the participant agent trust, agents that participant agent think that consider him as credible, friendship agents and enemy agents.



Although our model is based on the OCC model we think that with the inclusion of mood we can surpass one of the major critics that usually is pointed to this model, the fact that OCC model does not handle treatment of past interactions, past emotions.

## 5 Conclusion

More and more organizational decisions are taken by groups of people distributed in time and space. It is also accepted that the emotional state of an individual affects its decision and when he is taking part of a group decision he will influence both the emotional state of others members and group decisions.

In this paper it was briefly presented *WebMeeting* a ubiquitous group decision support system, but its main goal was the presentation of an agent based simulation model to group decision. The presented model incorporates the agents' emotions and mood in the decision making process. The agent emotions and mood affect the selection of arguments to send to other agents, as well as, the evaluation of the received arguments. Agents' individual emotions and mood are affected by the process of mood contagion.

Future work includes the implementation, validation and consequent refinement of the model. The inclusion of this model as a component of *WebMeeting* is also being considered. In that case a participant in a real ubiquitous group decision meeting, supported by *WebMeeting*, will use this model for instance to simulate the other participants and to preview its behaviour.

## References

1. Adamatti, D. and Bazzan, A.: AFRODITE – A Framework for Simulation of Agents with Emotions. ABS 2003 – Agent Based Simulation 2003. Montpellier, France, 28-30 April, (2003)
2. Bates, J.: The role of emotion in believable agents. Communications of the ACM, Special Issue on Agents, July (1994)
3. DeSanctis, G. and Gallupe, R. B.: Group Decision Support Systems - A New Frontier. Database Vol. 16 No. 1 (1985) 3-10
4. Elliot, C.: The Affective Reasoner A process model of emotions in a multi-agent systems. PhD dissertation. Northwestern University, USA, (1992)
5. El-Nasr, M.; Yen, J.; Ioerger, T.R.: FLAME -Fuzzy Logic Adaptive Model of Emotions. Autonomous Agents and Multi-agent systems, Vol.3 (2000) 217-257
6. Grudin, J.: Group Dynamics and Ubiquitous computing. Communications of the ACM, vol 45 No. 12 (2002)
7. Huber, G. P. : Group decision support systems as aids in the use of structured group management techniques. Proc. of second international conference on decision support systems, San Francisco, (1982) in C. W. Holsapple, A. B. Whinston, Decision support systems: a knowledge-based approach (Thomson Learning, inc, 2001)
8. Huber, G. P: Issues in the design of group decision support systems, Mis Quarterly, Vol. 3 No. 8 (1984).
9. Ito, T.; Shintani, T.: Persuasion among agents: An approach to implementing a group Decision Support System based on multi-agent negotiation. Proceedings of the 5<sup>th</sup> International joint Conference on Artificial Intelligence (1997)

10. Karacapilidis, N.; Papadias, D.: Computer supported argumentation and collaborative decision making: The Hermes system, *Information Systems*, Vol. 26 No. 4 (2001) 259-277
11. Kudenko, D.; Bauer, M.; Dengler, D.: Group decision making through mediated discussions. Proceedings of the tenth International conference on user modelling (UM'03) (2003)
12. Lewis, L.F.: Facilitator: A microcomputer decision support systems for small groups, Ph. D. dissertation, University of Louisville, 1982. in C. W. Holsapple, A. B. Winston, *Decision support systems: a knowledge-based approach* (Thomson Learning, inc, 2001).
13. Marreiros, G.; Ramos, C. and Neves, J.: Emotion and Group Decision Making in Artificial Intelligence. Proceedings of InterSymp 2005 17th International Conference on System Research, Informatics & Cybernetics - Special Focus Symposium on Cognitive, Emotive and Ethical Aspects of Decision-Making in Humans and in AI, Baden-Baden, Germany (2005)
14. Marreiros, G.; Ramos, C. and Neves, J.: Modelling group decision meeting participants with an Agent-based approach. Selected for publication in an upcoming special issue of the *International Journal of Engineering Intelligent Systems* (2005)
15. Marreiros, G.; Santos, R.; Ramos, C. and Neves, J.: Agent Based Simulation for Group Formation. SCS-ESM 2005 19th European Simulation Multi-Conference, Riga, Latvia (2005) 521-526
16. Marreiros, G.; Sousa, J.P. and Ramos, C.: WebMeeting - A Group Decision Support System for Multi-criteria Decision Problems. *International Conference on Knowledge Engineering and Decision Support*, Porto, Portugal ICKEDS04 (2004) 63-70
17. Neumann, R. and Strack, F.: Mood contagion: The automatic transfer of mood between persons; *Journal of Personality and Social Psychology*, Vol. 79 (2000) 211-223
18. Nunamaker, J.F. et al.: Lessons from a dozen years of group support systems research: A discussion of lab and field findings. *Journal of Management Information Systems*, Vol. 13 No. 3 (1997).
19. Ortony, A.: On making believable emotional agents believable. In R. P. Trappl, P. (Ed.), *Emotions in humans and artefacts*. Cambridge: MIT Press (2003)
20. Ortony, A.; Clore, G.L.; Collins, A.: *The cognitive structure of emotions*. Cambridge: Cambridge University Press (1988)
21. Picard, R. : *Affective Computing*; MIT Press, Cambridge, MA (1997)
22. Roseman, I.; Spindel, M.; Jose, P.: Appraisals of emotion-eliciting events: Testing a theory of discrete emotions. *Journal of Personality and Social Psychology*, Vol. 59, (1990)

# A Multi-agent Software Platform Accommodating Location-Awareness for Smart Space

Hongliang Gu, Yuanchun Shi, Guangyou Xu, and Yu Chen

Computer Science Department, Tsinghua University, Beijing 100084, P.R. China  
ghl02@mails.tsinghua.edu.cn, shiyc@tsinghua.edu.cn,  
xgy-dcs@mail.tsinghua.edu.cn, yuchen@tsinghua.edu.cn

**Abstract.** Software Platform is a middleware component of Smart Space to coordinate and manage all modules. Location-awareness is a common feature of many modules. Current several typical methods for distributed systems can hardly be competent for both the role of Software Platform and accommodating location-awareness simultaneously. Aiming at this, we present our method: SLAP (Smart Location-awareness-Accommodating Platform). The method, on the basis of OAA (Open Agent Architecture), adopts such new technologies as *Poll-Ack mechanism*, *dual-central coupling model* and *hybrid architecture*. Consequently it not only reserves the advantages of OAA to coordinate multi-modal modules efficiently and flexibly, but also accommodates location-aware computing well.

## 1 Introduction

Smart Space [1] (or Intelligent Environment) is a working environment integrated with numerous distributed software and hardware, including multi-modal modules and positioning sensors, which is also a system intensively applying pervasive/ubiquitous computing technologies. The Software Platform (also called Software Infrastructure), working as a middleware between OS (Operation System) and application modules, is a fundamental component of Smart Space to coordinate and managing all hardware and software modules.

Nowadays location-awareness is becoming an indispensable characteristic of most modules in Smart Space, which brings about a research field: location-aware computing. In our project, Smart Classroom [2] (a Smart Space on tele-education), location-awareness means that applications or services can modify their own behaviors unobtrusively or non-intrusively to adapt to users' purpose, according to the location (or spatial relationship) of located-objects [3] (including applications or service).

Both accommodating location-aware computing and adapting to Smart Space give the Software Platform dual challenges, which are just all the necessity of Smart Classroom. However, current several representative methods for tradition distributed systems, e.g. DCOM, CORBA, Metagluce and OAA (Open Agent Architecture) etc, can not give both needs a satisfying solution simultaneously. Aiming at this, we present our method: SLAP, a system with our improvement on OAA, which it not only efficiently coordinates and manages all modules according to the demands of Smart Space, but also accommodates location-aware computing very well.

The contents below are as follows: Section 2 discusses the demands of Smart Space on Software Platform. Section 3 introduces the requirements of location-awareness and the deficiencies of OAA. Section 4 presents our improvement's key technologies of. Section 5 presents the architecture and primitives of SLAP. Section 6 elaborates on the experiments. And section 7 concludes this paper.

## 2 Criterion and Selection of Software Platform

### 2.1 Demands of Smart Space on Software Platform

As far as the fact that Smart Space consists of various computing and communication units is concerned, Smart Space is a distributed system in some sense. However, it has some special features different from the normal distributed systems:

1. autonomy and independency

The modules in Smart Space are more autonomous and independent than those in distributed systems. For example, most modules in Smart Space can run or expire independently, which are not in a certain module's control and do not comply with other modules' assignment at all.

2. loose-coupling

A Smart Space system is very dynamic. Modules are restarted or moved to different hosts and System configurations change time to time. The loose coupling of modules will help to cope with this nature of Smart Space, as well as to resile from failure.

3. lightweight

As an underlying component, the Software Platform is to run on the various units in Smart Space which have various abilities of computing and communication spanning from mainframe computers to embedded systems, and to be used by the various module's developers who have uneven IT backgrounds. Thus, the feature of lightweight helps the Software Platform to accommodate various units, and to give various users a facile and simple interface. The light-weighted Software Platform only provides some key services and commits other complex functions to the applications in manner of the end-to-end implementation.

The items above are almost the common demands for all modules of Smart Space, especially for the multi-modal modules.

### 2.2 Selection of Software Platform

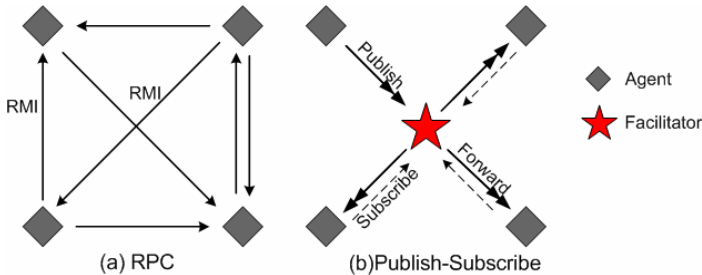
The Smart Space's features mentioned above are the criterion to select proper framework model for Software Platform. Currently, the representative methods for distributed systems can roughly be divided into two categories: Distributed Component Model (DCM), and Multi-Agent System (MAS).

In essence, DCM model is to encapsulate modules into objects (though someone argue that component is slightly different from object), which abstractly represents the states and behaviors' implementation (also called properties and methods) of modules. The representative DCM models include DCOM, CORBA and EJB etc. In DCM model, there must be a centralized thread of application logic which decides which objects to be invoked (used) and when to invoke (use). However, this premise is diffi-

cult to be met in Smart Space, due to the modules' autonomy and independency. For example, in Smart Classroom, a laser-pen-tracking module continuously tracks the position of laser point, which is a projecting point on Smart Board (a large-sized touch screen) corresponding to users' gesture, while a speech-recognition module keeps recognizing the user's voice. In the example, a clear centralized control logic is difficult to be picked up. Instead, there are two parallel application logics simultaneously.

In contrast, MAS model encapsulates each module into an agent, which not only has the same behaviors' implementation as an object, but also owns itself activation logic, executing process and purpose. That is, according to its environment, an agent can itself decide what to do and how to do, which an object can hardly achieve. Thus, in MAS model, the control logic of modules is decentralized, which is more flexible and fitter for Smart Space than that of DCM model. The typical MAS models include Metagluue [4], Hypergluue [5] and OAA [6],[7].

Besides those advantages, MAS model is usually more light-weighted than DCM model. Those representative DCM models, such as DCOM and CORBA, all own many complicated features, e.g. object set, transaction process and currency control etc. In view of those synthetic factors, we select MAS system instead of DCM model as the abstraction model of Software Platform. According to modules' coordination mode, MAS model is divided into two kinds: direct-coupled and meeting-oriented.



**Fig. 1.** The inter-module communication mode

The direct-coupled mode is also called RPC-like (Remote Procedure Call) mode. In this mode, each module must know other modules' definite reference (name or ID). As Fig. 1(a) shows, the module on the one side must know who the other side is, and furthermore the modules on the two sides must run at the same time. Both Metagluue and Hypergluue belong to this mode, in which the inter-agent communication is achieved by Java RMI. Undoubtedly, the direct-coupled mode is tight-coupling.

The meeting-oriented mode means the modules achieve the mutual coordination by broadcasting messages in a logic (virtual) meeting room. This mode's feature is that the modules needn't own others' references. The Publish-Subscribe mechanism, which OAA adopts, is typically meeting-oriented. In OAA, when an agent wants a certain kind of message, it will register the messages on a message center: *Facilitator*. This activity is called *subscribe message*, which is also called asking a question in OAA. And if an agent tends to send messages, it needn't know which agent and how many agents need those messages. What it does is only to send Facilitator the messages tagged with the name or category, and then Facilitator forwards all messages to

those agents who subscribe them, according to the messages' category and name. The agent's activity above is called *publish message*. The whole process is called "*delegated computing*" in term of OAA, which is skeletally shown in Fig.1 (b).

In comparison with the tight-coupling coordination mode of Metagluue, that of OAA is loose-coupling. Considering this factor, we prefer OAA to Metagluue and Hyperglue as a framework of Software Platform.

### 3 Deficiencies of OAA on Location-Awareness

#### 3.1 Requirements of Location-Awareness on Software Platform

In Smart Space, the location-aware computing system consists of three parts: location-aware applications, location server and position system. The position system of our project is Cricket V2.0 [8], in which each positioning unit (a PDA with Cricket Mote) knows its own geometric coordinate location and then sends its location to the location server by a wireless network. The location server, on the one hand, takes charge of storing and managing all units' location; on the other hand, provides the location-related services for the applications. In Smart Classroom, the location server adopts an implementation method called ASMod [9] to provide two kinds of service: query service and spatial event service. The former asks the applications' spatial query, which is like a SQL service; the latter tracks the varying of located-objects' spatial relationship to emit the relevant event notification.

To support the location-aware computing system, Software Platform encounters two new issues: one is how to efficiently organize the communication of position system, namely the communication between the location server (also an agent) and positioning agents (which correspond to positioning units); another is how to organize the communication between location-aware applications according to their locations (or spatial relationship) which is also called *location-based communication*. Unfortunately, neither of the issues is OAA competent for.

#### 3.2 Deficiencies of OAA on Supporting Location-Aware Computing

First, OAA does not excel at organizing the communication between the location server and positioning agents efficiently, due to its Publish-Subscribe mechanism. In the mechanism, when to publish messages and how many messages to publish only depend on the agent itself, which we call *free-publishing* characteristic. This characteristic adapts to such Smart Space's demands as modules' autonomy and independency and the system's loose coupling, meanwhile it also brings about two problems:

One problem is the difficulty in controlling the communication between the location server and positioning agents. In Smart Space, the location server usually needs to obtain the location from various positioning agents at various frequencies in different time according to its data's state, which is essentially the location server's data update policy. For example, in a time, if the location server infers that a positioning agent is moving quickly (maybe attached to a mobile person), it will get data from the agent twice per second. Likewise, in another time, if the location server infers the positioning agent seldom moves, it will get data from the agent only once per minute.

Another problem is that the disorderly contentions on the wireless network's channel increase, which results in the degradation of performance and throughput. Because each positioning agent publishes its data (namely location) only according to its own willing, despite the others and the location server's need, the disorderly contentions are inevitable, which will become more intensive with the increasing of the positioning agents' number and the frequency of publishing in each agent.

Secondly, OAA is also incompetent for organizing the location-based communication between applications. In Smart Space, much communication between agents is not constantly sustaining from beginning (subscribing) to end (unsubscribing), but varies according to their spatial relationship. The kind of communication, namely the location-based communication, is different from that of multi-modal modules, which OAA excels at. For example, when a PDA enters the service scope of Smart Board, the communication between the PDA agent and the Smart Board agent will emerge; and when the PDA leaves the service scope, the communication will also be broken off. Unfortunately, OAA is incompetent for the location-based communication. The cause is that neither Facilitator nor the source agents (which publish messages) cares the agents' location and changes their behaviors according to the varying of location.

## 4 Key Technologies of Our Improvement

Aiming at the deficiencies of OAA on supporting location-aware computing, we present our solution to Software Platform: SLAP (Smart Location-awareness-Accommodating Platform). Here we first introduce the Key technologies of SLAP, which are to solve the two issues brought by location-aware computing.

### 4.1 Poll-Ack Mechanism

Aiming at the incompetence of Publish-Subscribe mechanism for organizing position system communication, we present an appropriative inter-agent communication mechanism: Poll-Ack mechanism. This mechanism is described as follows:

As Fig. 2 illustrates, the communication consists of Poll-Ack cycles. And each cycle is initiated by a broadcast message from the location server, which is called *Poll*. A poll indicates which agent to publish its location. On receiving the *Poll*, the positioning agent indicated in the *Poll*, replies an acknowledgement message called *Ack* (including ID and location) in a fixed time. The location server stores all agents' location, and assigns poll number to each agent in the unit time according to the agent's velocity. An agent's velocity is its adjacent location difference divided by the interval

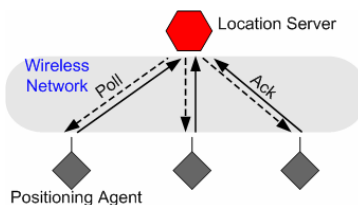


Fig. 2. The Poll-Ack mechanism

of its adjacent Ack. The higher velocity an agent is at, the more polls the location server assigns to it. Hence, not only this mechanism doesn't produce channel contention, but also it is a velocity-directed bandwidth assignment in some sense.

### 4.2 Dual-Central Coupling Model

Aiming at the incompetence of OAA for organizing location-based communication, we present the dual-central coupling structure and the Spatial-event-directed Publish-Subscribe mechanism.

In OAA there is a unique coupling center, Facilitator, to organize message communication. In contrast, in SLAP there are two coupling centers: LAMD (Location-Aware Message Dispatcher) and LocServ. The former provides the analogous function of Facilitator, and the latter plays the role of location server. LocServ has a component, *Spatial Event Generator*, which tracks the agent's moving and translates location into spatial events. LAMD owns a dispatching engine, *Forward-Valve*, which decides messages whether to forward indeed according to the event notification from LocServ. The structure of two coupling centers is shown in Fig. 3.

The dual-central coupling structure adopts a new communication mechanism called Spatial-event-directed Publish-Subscribe. The mechanism is based on Publish-Subscribe with some modification. The modification is as follows:

1. When an agent subscribes a kind of messages, it is demanded to submit a spatial condition of the messages to LocServ at the same time. The spatial condition indicates the premise the communication needs, and the premise is express as a spatial relationship, such as, the publisher's location must be contained in the subscriber's scope. The step is called *spatial condition's customization*.
2. LocServ keeps on obtaining the latest location of all agents from the position system (namely tracking agents' moving), and judges whether the spatial conditions are met by its Spatial Event Generator. When the spatial conditions become met or unmet, LocServ sends LAMD the event notifications: *message-forward-enable* or *message-forward-disable*.
3. According to the notifications, LAMD will decide whether to forward the subscribed messages to the subscriber (agents).

If an agent wants the received messages to be irrelevant to the location, it submits a command to LocServ to abolish the spatial condition of messages. The whole process of this mechanism is shown in Fig. 4.

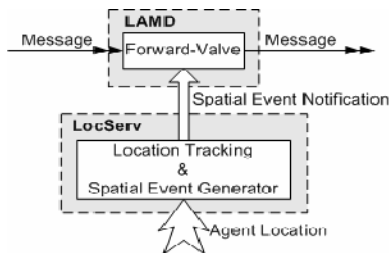


Fig. 3. The structure of two coupling centers



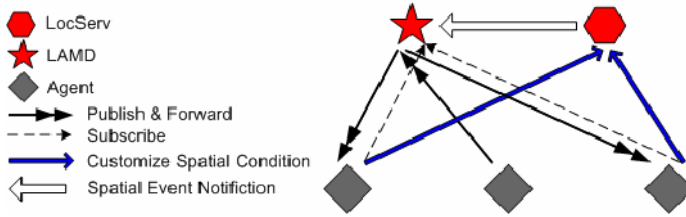


Fig. 4. The spatial-event-directed publish-subscribe mechanism

## 5 The Architecture of SLAP

As a Software Platform, SLAP is a middleware between OS (Operation System) and applications (agents). An overview of SLAP is shown in Fig. 5.

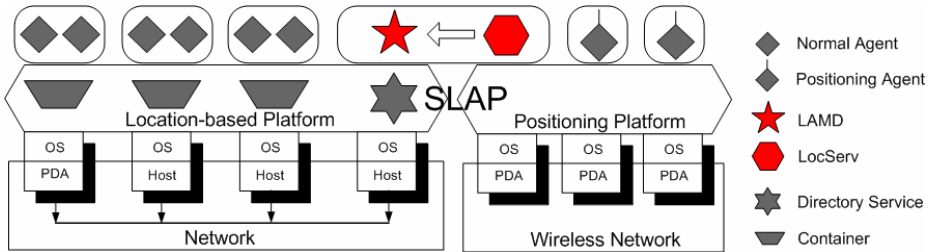


Fig. 5. The architecture of SLAP

SLAP is a *hybrid architecture* composed of two parts, which respectively correspond to two kinds of communication environment. The right part, *positioning platform*, is used for the position system to coordinate LocServ and the positioning agents, which adopts the Poll-Ack communication mechanism. And the left part, *location-based platform*, is to organize the location-based communication between agents, which adopts the dual-central coupling model. The host containing LAMD and LocServ spans two network environments: one connects to the position system’s network (wireless network), another connects to the network all normal agents share. To enhance some functions of SLAP, we add in some components that OAA doesn’t own. For example, the *containers*, acting as mediators under the agent layer, are to shield heterogeneous OS and accommodating different developing languages, such as C++ and Java. The *Directory Service* is used for the service’s discovery.

## 6 Performance Analysis

To evaluate the performance of SLAP, we compare the Poll-Ack mechanism (which SLAP adopts) with the Publish-Subscribe mechanism (which OAA adopts) on the communication efficiency of position system. Define:

$T_L$  = Average time for a positioning agent to calculate its location

$D_L$  = Transmission duration of a location message (which is in the form of Ack message in the Poll-Ack mechanism)

$D_p$  = Transmission duration of a poll message

Now we first investigate the performance of Publish-Subscribe mechanism. Providing a positioning agent publishes its location message at once after calculating its location, the probability that the positioning agent publishes the location:

$$p = \frac{D_L}{T_L} \quad (1)$$

For a successful publishing exactly one of  $n$  positioning agents should be publishing at a given time. Hence the probability that only one given positioning agent is publishing at a particular time:

$$P_1 = p(1-p)^{n-1} \quad (2)$$

When there are  $n$  positioning agents, the probability that exactly one positioning agent is publishing at a given time is the channel utilization of wireless network  $U$ .

$$U = np(1-p)^{n-1} \quad (3)$$

For maximum utilization of publish-subscribe mechanism, there exists:

$$\frac{dU}{dp} = n(1-p)^{n-1} - np(n-1)(1-p)^{n-2} = 0 \Rightarrow p = \frac{1}{n} \Rightarrow T_L = nD_L \quad (4)$$

Hence, in the case above, the optimum utilization of Publish-Subscribe mechanism:

$$U_{o_{p-s}} = \left(1 - \frac{1}{n}\right)^{n-1} \quad (5)$$

As for the Poll-Ack mechanism, the channel is occupied by Polls and Acks in turn. Hence, the channel utilization  $U'$  is:

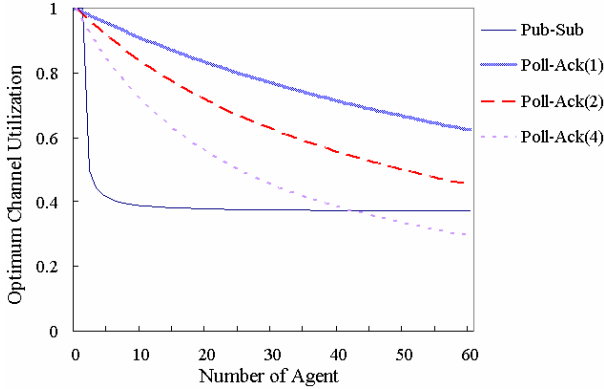
$$U' = \frac{D_L}{D_L + D_p} \quad (6)$$

Because the equation (4) exists in the case of channel's maximum utilization, the optimum utilization of Poll-Ack mechanism  $U_{o_{p-a}}$  is:

$$U_{o_{p-a}} = \frac{T_L}{T_L + nD_p} \quad (7)$$

As for a given position system, the average time of calculating location  $T_L$  is fixed, which only depends on the hardware's intrinsic functionality. In contrast, the poll's transmission duration  $D_p$  is determined by the concrete Poll-Ack mechanism.

Both Publish-Subscribe mechanism and Poll-Ack mechanism are simulated using the ns-2 network simulator with suitable extensions [10], which is guided by the CMU wireless extensions. In the simulation experiment,  $T_L$  is set to 100ms,  $D_p$  is set to 1ms, 2ms and 4ms, which are corresponding to the curve Poll-Ack (1), (2) and (4) in Fig. 6 respectively. And the performance of Publish-Subscribe mechanism is labeled by the curve Pub-Sub in Fig. 6.



**Fig. 6.** The communication performance of SLAP versus OAA in position system

As Fig. 6 shows, in most cases, the optimum channel utilization of the Poll-Ack mechanism is superior to that of the Publish-Subscribe mechanism. The few exceptional cases occur on Poll-Ack (4) with the agent number of about 45, where the total transmission duration of polls (180ms) is greater than the average positioning time (100ms) by far. These cases are very extreme, which rarely appears in practice. Another trend seen from Fig. 6 is that, the smaller the poll's transmission duration is, the larger improvement of channel utilization the Poll-Ack mechanism achieves on the Publish-Subscribe mechanism.

## 7 Conclusion

On the one hand, as a Software Platform for Smart Space, being based on OAA, SLAP reserves the main characteristics of OAA, a loose-coupling multi-agent system. Those characteristics conform to Smart Space's demands on Software Platform better than other distributed system methods, which highly ensure modules' autonomy and independency, inter-module loose-coupling and system's lightweight. Hence, as far as accommodating Smart Space is concerned, SLAP, as well as OAA, is an excellent Software Platform, especially for coordinating most multi-modal modules.

On the other hand, SLAP overcomes the shortcomings of OAA on accommodating location-awareness. By introducing in the dual-central coupling model, SLAP realizes inter-agent location-based communication that OAA used to not be able to provide. And by introducing the Poll-Ack communication mechanism into position system,

SLAP achieves higher channel utilization and more efficient communication performance than OAA. Thus SLAP not only is competent for Software Platform of Smart Space, but also accommodates location-aware computing well.

## References

1. <http://www.nist.gov/smartspace/>
2. Y. C., Shi, et al.: The smart classroom: merging technologies for seamless tele-education. *Pervasive Computing*, IEEE press, Vol 2, No 2, 2003, pp. 47-55
3. B. Schilit, N. Adams, and R. Want: Context-aware computing applications. *IEEE Workshop on Mobile Computing Systems and Applications*, IEEE CS Press, 1995, pp. 85-90
4. M.H. Coen, B. Phillips, N. Warshawsky, et al.: Meeting the computational needs of intelligent environments: The Metaglow system. *Proc 1st International Workshop Managing Interactions in Smart Environments (MANSE'99)*, 1999, pp.210-213
5. Peters S, Look G, Quigley K.: *Hyperglue: Designing High-Level Agent Communication for Distributed Applications*. Technical Report, Laboratory of CS and AI (CSAIL), Massachusetts Institute of Technology, 2002.
6. SRI., OAA web site: <http://www.ai.sri.com/~oaa>
7. Adam Cheyer, David Martin: *The Open Agent Architecture*. *Autonomous Agents and Multi-Agent Systems*, Kluwer Academic Publisher, Vol 4, No 1-2, 2001, pp.143-148
8. Adam Smith, Hari Balakrishnan, Michel Goraczko, Nissanka Priyantha: Tracking Moving Devices with the Cricket Location System. *Proc 2nd International conference on Mobile systems, applications, and services(MobiSys'04)*, 2004, pp.190-202
9. Hongliang Gu, et al.: A core model supporting location-aware computing in Smart Classroom, *Proc 4th International Conference on Web-based Learning*, 2005, pp.1-13
10. NS-2 network simulator. <http://www.isi.edu/nsnam/ns/>

# Context-Aware Ubiquitous Data Mining Based Agent Model for Intersection Safety\*

Flora Dilys Salim<sup>1</sup>, Shonali Krishnaswamy<sup>1</sup>, Seng Wai Loke<sup>1</sup>,  
and Andry Rakotonirainy<sup>2</sup>

<sup>1</sup> Caulfield School of Information Technology, Monash University,  
900 Dandenong Road, Caulfield East, VIC 3145, Australia  
{Flora.Salim, Shonali.Krishnaswamy,  
Seng.Loke}@infotech.monash.edu.au

<sup>2</sup> Centre for Accident Research and Road Safety Queensland,  
Queensland University of Technology, Beams Road, Carseldine, QLD 4034, Australia  
r.andry@qut.edu.au

**Abstract.** In USA, 2002, approximately 3.2 million intersection-related crashes occurred, corresponding to 50 percent of all reported crashes. In Japan, more than 58 percent of all traffic crashes occur at intersections. With the advances in Intelligent Transportation Systems, such as off-the-shelf and in-vehicle sensor technology, wireless communication and ubiquitous computing research, safety of intersection environments can be improved. This research aims to investigate an integration of intelligent software agents and ubiquitous data stream mining, for a novel context-aware framework that is able to: (1) monitor an intersection to learn for patterns of collisions and factors leading to a collision; (2) learn to recognize potential hazards in intersections from information communicated by road infrastructures, approaching and passing vehicles, and external entities; (3) warn particular threatened vehicles that are approaching the intersection by communicating directly to the in-vehicle system.

## 1 Background

In spite of the advancement of state-of-the-art technologies being implemented in vehicles and on the road over the years, the annual toll of human loss caused by intersection crashes has not significantly changed in more than 25 years, regardless of improved intersection design and more sophisticated ITS technology [21]. Intersections are among the most dangerous locations on U.S. roads [7]. In 2002, USA, approximately 3.2 million intersection-related crashes occurred, corresponding to 50 percent of all reported crashes. 9,612 fatalities (22 percent of total fatalities) [21] and roughly 1.5 million injuries and 3 million crashes took place at or within an intersection [22]. Yearly, 27 percent of the crashes in the United States take place at intersections [7]. In Japan, more than 58 percent of all traffic crashes occur at intersections. Intersections-

---

\* The work reported in this paper has been funded in part by the Co-operative Research Centre Programme through the Australian Government's Department of Education, Science and Training.

related fatalities in Japan are about 30 percent of all Japanese traffic accidents, and those fatal crashes mainly happen at intersections without traffic signals [7].

The complexity of intersections is due to various characteristics of intersections [1, 7, 19], which are as follows: different intersection shapes, number of intersection legs, signalized/ unsignalized, traffic volume, rural / urban setting, types of vehicles using the intersection, various average traffic speed, median width, road turn types, and number of lanes. From those characteristics that pertain to intersection collisions, a driving assistance system for intersection is highly needed, particularly one that is able to warn driver for potential threats or collisions. Given the uniqueness of each intersection, an intelligent system for intersection safety should be able to adapt to different characteristics of an intersection [19].

The advances in sensor technology and the need for intelligence, dynamicity, and adaptability in ITS have motivated the research of Context-Awareness, Multiagent Systems, and Data Mining for Intelligent Transportation Systems as discussed in Section 2. Section 3 discusses the model we propose to address the issues of intersection safety. Section 4 concludes the paper and outlines future work of the project.

## 2 Related Work

Subsection 2.1 reviews existing research projects in intelligent software systems, such as Context-Awareness, Multiagent Systems, and Data Mining, which have been utilized to advance Intelligent Transportation Systems. Subsection 2.2 discusses the existing approaches in intersection collision warning and/or avoidance systems.

### 2.1 Intelligent Software Systems

Context-aware applications observe the “*who*’s, *where*’s, *when*’s, and *what*’s” of entities and use this information to find out “*why*” a situation is happening [2]. With the availability of context information, an application can then use it to adapt to environment changes. The research areas of context-awareness in ITS include smart autonomous cars [17, 18] and traffic monitoring [11].

An agent is autonomous intelligent program acting on behalf of the user [24]. A multiagent system (MAS) is a collection of agents that communicate with each other and work together to achieve common goals with a certain measure of reactivity and/or reasoning [24]. There have been considerable ITS projects using the notion of agents, such as for controlling and managing traffic in intersections [3, 6, 10].

Given that there are considerable amount of data from the in-vehicles and roadside sensors, clearly, it is essential to make sense of the sensors data. Data mining is the development of methods and techniques to gain knowledge from data by pattern discovery and extraction [4]. Data analysis techniques are necessary for processing information both on roadside and in vehicle situations [16]. However, data mining and machine learning techniques require high computational resources as knowledge is discovered from the analysis of huge data storages. Learning from data streams in ubiquitous environment is enabled by Ubiquitous Data Mining (UDM), which is the analysis of data streams to discover useful knowledge on mobile, embedded, and ubiquitous devices [9]. UDM have been used to monitor vehicle’s health and driver’s characteristics in moving vehicles [13] and to identify drunk-driving behavior [12].

The above mentioned technology in ubiquitous computing enables more sophisticated ITS applications. However, after reviewing those research projects, none has addressed a holistic approach for intersection safety.

## 2.2 Intersection Collision Warning and/or Avoidance Systems

Intersection collision warning and avoidance systems are categorized as either *vehicle-based*, *infrastructure-only* or as *infrastructure vehicle cooperative* [5, 22]. *Vehicle-based systems* rely only on in-vehicle sensors, processors, and interface to detect threats and produce warnings [22]. *Infrastructure-only systems* rely only on roadside warning devices to inform drivers [5]. *Cooperative systems* communicate information straight to vehicles and drivers. The main advantage of cooperative systems rests in their potential to improve the interface to the driver, and thus to almost guarantee that a warning is received.

Existing Intersection Collision Warning Systems as those described in [5, 8, 19, 20, 23] are still infrastructure-only system, and are limited in certain aspects:

1. Warning messages are less effective as they are only displayed on the roadside.
2. There is no communication means that exists between road infrastructure and vehicles, and therefore, no exchange of useful information between them.
3. Information about intersection might not be comprehensive as the only data source is roadside sensors.
4. The systems are mostly reactive. Although reactive trait is required; however, deliberative reasoning aspect can supplement and enhance these systems.
5. Each system is built for a particular intersection or an intersection type, and therefore each application requires a field study on that intersection.

Vehicle-based intersection collision warning systems [15] are fairly effective for a single vehicle. However, in an intersection, a cooperative system is a preferred solution as it is very important to communicate foreseen threats to other vehicles.

Research initiatives in developing cooperative system for intersection safety such as [14, 22] have recently commenced. However, these projects do not mention the techniques to discover crash patterns and pre-crash behavior associations, which are essential to detecting and reacting to potential threats. A generic framework that is able to automatically adapt to various types of intersections is also required for efficiency of deployment; however, these projects have not addressed this issue.

There is a project that uses multiagent system for intersection collision warning system [22]; however, it only implements vehicle-to-vehicle cooperation for intersection safety. Threat detection relies on information (location, velocity, acceleration) shared by other vehicles. Useful information from external sources such as the infrastructure and environment are not incorporated. Another limitation is that the agent architecture is reactive; there is no learning to gain new knowledge that can improve the system.

Therefore we suggest an integration of multi-agent systems and ubiquitous data mining notions to a hybrid intersection safety model that can be applied to any intersection. The elucidation and model of our approach is described in the Section 3.

### 3 Proposed Model

Subsection 3.1 outlines the requirements of the model for intersection safety management. Subsection 3.2 explains our model to answer those requirements.

#### 3.1 Model Requirements

There is a need for a cooperative intersection collision warning and avoidance system that addresses the following challenges:

1. An intersection safety model that is able to detect high risk situations and foresee threats in particular intersections is required. Given that there is considerable amount of sensor data in cars and infrastructures, there is an opportunity to reason and use this data to develop patterns and associations that can help in better understanding of high risk situations and behaviors that lead to crashes. While current systems tend to be reactive to situations, there is increased recognition [3, 14, 22] that reasoning and learning can be integrated to supplement reactivity.
2. As each intersection is unique, the profile of high risk situations in one intersection is different from another, therefore, a generic model that is able to adapt to particular intersections over a period of time is required. Each system in different intersections should have a knowledge that is applicable only within its locality. This knowledge is gained through reasoning and learning. Hence, this approach alleviates the inefficiency of the current method of developing different intersection collision warning and avoidance systems for different intersections [1, 7, 19].
3. There is a necessity for exchange of information and knowledge between intersection infrastructure and vehicles and also for vehicle-to-vehicle communication. This is due to the need for a comprehensive understanding of a particular intersection so that the system is able to act or respond better to a hazardous situation.

This research aims to investigate an integration of intelligent software agents, ubiquitous data stream mining, for a novel context-aware framework that is able to:

1. monitor an intersection to learn for patterns of collisions and factors leading to a collision using ubiquitous data stream mining;
2. learn to recognize potential hazards in intersections from information communicated by road infrastructures, approaching and passing vehicles, and external entities using a layered agent architecture;
3. warn particular threatened vehicles that are in the intersection by communicating directly to the in-vehicle system with multi-agent communication principles.

The goal is feasible due to the recent advances in ITS sensor technology that allows real-time data from in-vehicle and traffic sensors to become more accessible.

#### 3.2 Model Description

This research brings together Multi-Agent Systems with Ubiquitous Data Mining to develop a context-aware model that addresses for cooperative intersection collision warning and avoidance systems.

Multiagent technology is very fitting for coordination of entities in intersections. The abstraction of independent, autonomous entities that are able to communicate with



other entities and make independent decisions maps eminently to the situation of an on-road scenario. Each entity can be represented by an intelligent agent. Communication among those entities is made possible through agent communication language. Accordingly, we need to decide on which agent architecture is the most appropriate to answer the challenges in the Section 1. According to [24], there are four classifications of agents based on their architectures: logic, reactive, BDI, and layered agents. As agent's layered architecture is designed for balance of mutual effectiveness of reactivity and reasoning, thus we view it as appropriate to adopt this architecture for the basis of the model of agents for intersection safety system. Such model allows retaining the element of reactivity while incorporating the potential to reason and learning.

The question now remains as to how the reasoning and learning is accomplished. We view Ubiquitous Data Mining (UDM) as suitable in this context. A system that is deployed to continuously monitor an intersection must necessarily be able to operate in a ubiquitous resource-constrained environment. The information delivered to the systems will be from a myriad of sensors that continuously and rapidly stream data to the systems. Given this content, it is evident that UDM is a suitable option and one that can facilitate incremental learning. The question remains that while the general principals of UDM are appropriate for our research, the specifics and modalities of the learning process and the algorithms suited to this application need to be investigated and developed as part of this research.

Therefore, the model we propose is: *A context-aware multi-agent framework with an integration of layered agent architecture and ubiquitous data mining for intersection safety*. The subsection 3.2.1 discusses the internal model of agents, while the subsection 3.2.2 discusses the interaction model of our multiagent system.

### 3.2.1 Agent Model

For each agent in the framework, we propose a novel hybrid agent model: Ubiquitous Data Mining based Layered Agent (UDMLA), as displayed in Figure 1.

The theoretical model consists of three layers, which are described as follows:

1. Reactive layer as the bottom layer. It has sensors, communication components, and actuators that accept sensory data input and generate responses. It performs information exchanges with other agents or external parties and performs the task of issuing notifications. Reactive layer possesses knowledge based rules for generating actions or responses. The characteristics of the knowledge in this layer are stable (unchanging for an extended period of time) and highly reliable or have high levels of confidence.
2. Training layer is intended to test new knowledge from the higher layer. Data received from reactive layer are passed to the higher layer for reasoning. This layer is designed to train untested knowledge that is passed from reasoning layer by data mining techniques for training datasets, solve conflict in untested knowledge by confidence measurement, recognize failures and learn from it by passing the information back to the reasoning layer. This layer possesses knowledge with moderate confidence as the knowledge still needs to be tested. When this knowledge has reached acceptable levels of stability or confidence, it is passed to the reactive layer for initiating actions based on events that conform to these patterns.
3. Reasoning layer contains UDM algorithm that extracts information from streams of data to recognize new knowledge such as in form of patterns and associations.

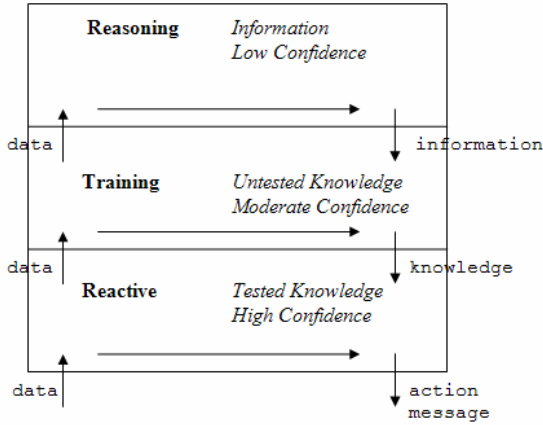


Fig. 1. UDM based Layered Agent (UDMLA) model

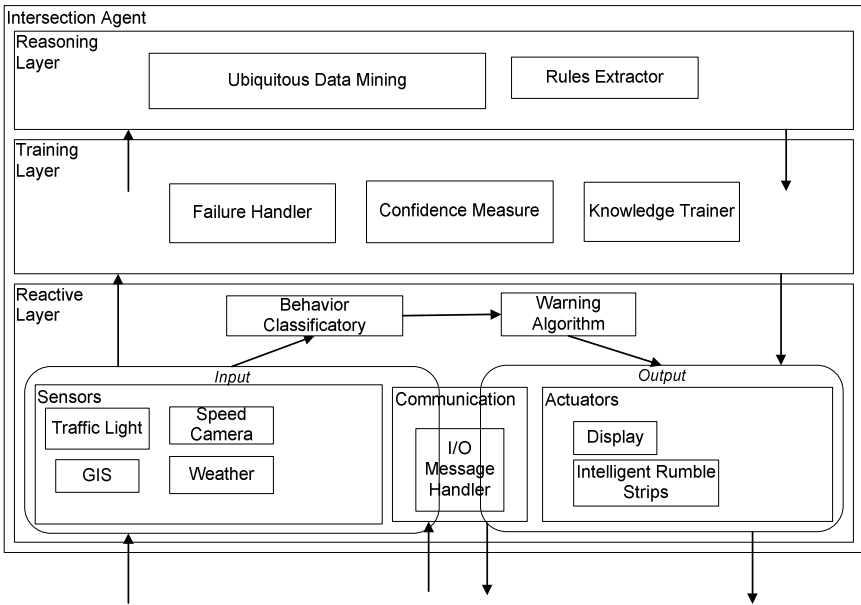


Fig. 2. The internal architecture of an intersection agent

Each layer has a confidence measure to check whether data entering the layer can be treated within certain levels of confidence for specific purposes such as for generating actions or training; otherwise, data will be passed into higher layers for reasoning. This approach facilitates knowledge evolution within the layers of the agent; hence, the agent is improving its intelligence over a period of time.

To our knowledge, a model of intelligent agent architecture that accommodates Ubiquitous Data Mining is novel. The UDMLA model is applicable to other applica-

tion domains that require reactivity along with deliberation to cope with a fast changing environment.

Figure 2 shows an application of UDMLA model for a single intersection agent. Input to the reactive layer of an intersection agent can come from sensory inputs and also from different sources, such as from vehicle agents and external parties such as traffic bureau. The input data is checked against behavior classificatory to be validated whether it falls into one of the dangerous behavior categories. If it does, the data will be passed on to warning algorithm that will take an action depends on the rate of danger a situation carries.

Every input to the reactive layer is also passed into the next upper layer, which is the training layer. The training layer assesses the input and remeasures the confidence of knowledge by calculating and comparing the number of valid and invalid matched data items. For example, if knowledge to be tested states that a car that a travel approaching the vicinity is making a direct left turn without first yielding right-of-way, and another car from the opposite side of intersection, with the distance less than 25 meters, is approaching with the average speed of 50 km/h, crash will happen. In this case, the crash will likely occur in 1.8 second ( $3600 \text{ seconds} \div (50000 \text{ m} \div 25 \text{ m})$ ). Say that this knowledge has 3 valid occurrences out of 4 total occurrences (75% confidence). A new data item that falls within the same situation adds the confidence of the knowledge to be 4 out of 5 (80% confidence). A confidence threshold is given to this layer, that before a knowledge can be passed into the reactive layer it must reach a certain level of confidence, for example 90% confidence. If there is a failure in warning relevant vehicles (i.e. crash happens), failure handler will store the case and test next relevant data items whether the correct rule is the negation of the current rule or is a fuzzy rule derived from both the current rule and its negation.

The top layer is the reasoning layer where all input data are being learned and studied by Ubiquitous Data Mining techniques to find patterns of intersection crashes and dangerous situations and driver behaviors that lead to each crash category. Rules for classifying situations are also being extracted here after clusters of crash patterns are found, so that dangerous situations can be detected instantly. Every new rule extracted is transferred to the training layer to be tested with new data items over a period of time.

One example of a scenario that is examined by an intersection agent is a situation of a small size car that is approaching the intersection with the speed of 40 – 60 km/h without decelerating to beat the yellow traffic light before it turns to red in 0.5 second. A near side-collision event occurs as a car from the other side of intersection suddenly puts on the brakes within the distance of 0.5 meter from the car that violates the red light signal. This event is then recorded with all the attributes to be clustered and classified by UDM algorithm. The clustering of UDM uses initial clusters depends on intersection types and crash patterns described by previous studies. For example, for a cross intersection [15], the initial clusters are: (1) across path turn; (2) perpendicular paths with no violation of traffic control; (3) perpendicular paths with violation of traffic control; (4) premature intersection entry scenario. Driving behaviors and attributes in each of the cluster will then be mapped against five stages of driving, which are “normal, warning, collision avoiding, collision imminent, and collision past” [24]. The warning algorithm treats each stage of driving differently by issuing different level of warning. The number of crash patterns will change according to the usage behaviors and characteristics of the intersection. Hence, the intersection agent is context-aware, and able to adapt to different kinds of intersections due to its learning capabilities.

Vehicle agents are using reactive agent architecture as immediate actions should be taken in response to warning messages from the intersection agent and possibly from other vehicle agents. Vehicle agents only carry knowledge that are tested and has a high level of confidence. This knowledge is communicated by the intersection agent. The multiagent interaction model used for the system is discussed in the next subsection 3.2.2.

### 3.2.2 Multiagent Interaction Model

The multi-agent system consists of a stationary agent in an intersection and also mobile agents in vehicles and is capable of discovering knowledge from streams of data from various sources such as sensors, traffic bureau and weather bureau. Multi-agent system will be applied on the whole intersection-vehicle system. Each vehicle will have at least one vehicle agent, and every intersection will have at least one stationary agent. These agents will then communicate and work together to achieve their common goals using their individual and shared knowledge delivered from ubiquitous data mining. As a result, the system will be more knowledgeable over periods of time. If a vehicle or a driver has unacceptable behaviors that will risk the other road users, mobile agents will warn the stationary agent in an adjacent intersection. If a danger for collision is foreseen by either the stationary agent at the intersection, warnings will be sent to all relevant vehicles. An agent that resides in each vehicle will then act accordingly to the warning message and also to the situation of the vehicle and driver. This architecture is general for all kinds of intersections, as each intersection will have its own set of localized knowledge. This is due to the different crash patterns that exist because of the situation difference, such as intersection shape, location, volume usage, and presence of different traffic signals. As a result, this infrastructure safety architecture is also a context-aware system that knows about its current situation and knows how to react and adapt to different situations. The intersection agent operates within its zone of influence.

A *zone of influence* is the spatial domain that determines the region of authority of an intersection agent to coordinate vehicle agents in the approaching and passing vehicles. Knowledge about an intersection that is possessed by an intersection agent is specific within the boundaries of the zone of influence. Once a vehicle enters a zone of influence, it broadcasts its sensor data to the intersection agent that resides in the zone of influence. The intersection agent will then transfer its knowledge about the intersection to the vehicle for the knowledge base of the vehicle agent's warning algorithm. Warnings are produced mainly from the vehicle agent when the agent detects the driver is executing dangerous driving maneuvers. However, warnings are also produced from the intersection agent and sent to relevant vehicles that are going to be affected, as at some situations where multiple cars are involved, it is only the intersection agent that is able to detect and analyze the situation well. In the intersection agent, the zone of influence is managed by I/O message handler in the reactive layer.

Our architecture for intersection collision warning and avoidance system enables vehicle-to-vehicle communication and vehicle-to-infrastructure communication via agent communication protocol. The necessity of applying data processing and analysis techniques to assess different situations in an intersection is satisfied by having ubiquitous data mining that is learning from sensors information. Another benefit of this approach is that it is a scalable solution as there is an automatic localization to specific intersections.

## 4 Conclusion and Future Work

We have proposed Ubiquitous Data Mining based Layered (UDMLA) model for cooperative intersection-vehicle safety: an integration of layered agent architecture with ubiquitous data mining and context-awareness for intersection safety with the notion of support and confidence of data mining for knowledge evolution of an agent.

Our contribution to research in road safety is a generic intersection safety model that can adapt to specific intersections. We are currently implementing the UDMLA model on a computer based simulation.

## References

1. Arndt, O. K.: Relationship Between Unsignalised Intersection Geometry and Accident Rates, School of Civil Engineering, Queensland University of Technology, PhD Thesis (2003)
2. Dey, A. K. and Abowd, G. D.: Towards a Better Understanding of Context and Context-Awareness, 1st International Symposium on Handheld and Ubiquitous Computing, GVU Technical Report GIT-GVU-99-22 (1999)
3. Dresner, K. and Stone, P.: Multiagent Traffic Management: An Improved Intersection Control Mechanism, the Proceedings of The Fourth International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS '05), Utrecht, Netherlands (2005)
4. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P.: From Data Mining to Knowledge Discovery in Databases, *AI Magazine* Vol. 17, No. 3 (1996)
5. Ferlis, R. A.: Infrastructure Intersection Collision Avoidance, Intersection Safety Conference, Milwaukee, WI (2001)
6. France, J. and Ghorbani, A. A.: A Multi-Agent System for Optimizing Urban Traffic, Proc. of IEEE/WIC International Conference on Intelligent Agent Technology (IAT 2003), Halifax, Nova Scotia, Canada (2003)
7. Frye, C.: "International Cooperation to Prevent Collisions at Intersections", *Public Roads Magazine*, Vol. 65, No. 1, July–August 2001, Federal Highway Administration, USA (2001)
8. Funderburg, K. A.: "Update on Intelligent Vehicles and Intersections", *Public Roads Magazine*, Vol. 67, No. 4, January–February 2001, Federal Highway Administration, USA (2004)
9. Gaber, M. M., Krishnaswamy, S., and Zaslavsky, A.: Ubiquitous Data Stream Mining, Current Research and Future Directions Workshop, in conjunction with The Eighth Pacific-Asia Conference on Knowledge Discovery and Data Mining, Sydney, Australia (2004)
10. Gabric, T., Howden, N., Norling, E., Tidhar, G., and Sonenberg, E.: Agent-oriented design of a traffic flow control system, University of Melbourne Department of Computer Science Technical Report 94/24 (1994)
11. Harrington, A., and Cahill, V.: Route Profiling - Putting Context To Work, Proceedings of the 19th ACM Symposium on Applied Computing (SAC 2004), Nicosia, Cyprus (2004) 1567-1573
12. Horovitz, O., Gaber, M. M., and Krishnaswamy, S.: Making Sense of Ubiquitous Data Streams: A Fuzzy Logic Approach, to appear in the Proceedings of the 9th International Conference on Knowledge-based Intelligent Information & Engineering Systems 2005 (KES 2005), Melbourne, Australia (2005)
13. Kargupta, H., Bhargava, R., Liu, K., Powers, M., Blair, P., Bushra, S., Dull, J., Sarkar, K., Klein, M., Vasa, M. and Handy, D.: VEDAS: A Mobile and Distributed Data Stream Mining System for Real-Time Vehicle Monitoring, Proceedings of the SIAM International Data Mining Conference, Orlando (2004)

14. Lages, U.: INTERSAFE – New European Approach for Intersection Safety, funded by the European Commission in 6th Framework Program, 11th World Congress on ITS, Nagoya, Japan (2004)
15. Lloyd, M., Pierowicz, J., Jocoy, E., Pirson, B., Bittner, A.: Intersection Collision Avoidance Using Its Countermeasures. Task 9: Final Report: Intersection Collision Avoidance System Performance Guidelines, U. S. Department of Transportation, National Highway Traffic Safety Administration (2000)
16. Miller, R., Huang, Q.: An Adaptive Peer-to-Peer Collision Warning System, Vehicular Technology Conference (VTC), Birmingham, Alabama (2002)
17. Oliver, N. and Pentland, A.: Graphical Models for Driver Behavior Recognition in a SmartCar, Proceedings of IEEE International Conference on Intelligent Vehicles, Detroit, Michigan (2000)
18. Sivaharan, T., Blair, G., Friday, A., Wu, M., Duran-Limon, H., Okanda, P., and Sørensen, C-F.: Cooperating Sentient Vehicles for Next Generation Automobiles, ACM MobiSys International Workshop on Applications of Mobile Embedded Systems, Boston (2004)
19. Stubbs, K., Arumugam, H., Masoud, O., McMillen, Veeraraghavan, H., Janardan, R., Papanikolopoulos, N.: A Real-Time Collision Warning System for Intersections, Proceedings of Intelligent Transportation Systems America, Minneapolis (2003)
20. U.S. Department of Transportation – Federal Highway Administration: Intersection Collision Warning System, April 1999, <http://www.tfhrc.gov/safety/pubs/99103.pdf> (1999)
21. U.S. Department of Transportation – Federal Highway Administration, Institute of Transportation Engineers: Intersection Safety Briefing Sheet, April 2004, <http://www.ite.org/library/IntersectionSafety/BreifingSheets.pdf> (2004)
22. U.S. Department of Transportation: Cooperative Intersection Collision Avoidance System, <http://www.its.dot.gov/initiatives/initiative2.htm> (2005)
23. Veeraraghavan, H., Masoud, O., and Papanikolopoulos, N.: Vision-based Monitoring of Intersections, Proceedings of IEEE Intelligent Transportation Systems Conference (2002)
24. Wooldridge, M.: “Intelligent Agents”, Multiagent systems: A modern approach to distributed artificial intelligence, Chapter 1, Weiss, G. (Ed.), The MIT Press (1999) 27 -77

# Development of Knowledge-Filtering Agent Along with User Context in Ubiquitous Environment

Takao Takenouchi<sup>1</sup>, Takahiro Kawamura<sup>2,3</sup>, and Akihiko Ohsuga<sup>2,3</sup>

<sup>1</sup> NEC Corporation, 2-11-5 Shibaura, Minato-ku, Tokyo, Japan  
takenouchi@bu.jp.nec.com

<sup>2</sup> The Graduate School of Information Systems,  
University of Electro-Communications, 1-5-1, Chofugaoka, Chofu-shi, Tokyo, Japan  
{kawamura, ohsuga}@maekawa.is.uec.ac.jp

<sup>3</sup> Research & Development Center, Toshiba Corp.,  
1, Komukai Toshiba-cho, Saiwai-ku, Kawasaki-shi, Kanagawa, Japan

**Abstract.** In this paper, we propose combination of Ubiquitous Computing and Semantic Web. Data and services will be annotated even in the ubiquitous devices, and should be connected to the web of the semantics near future. We call it Ubiquitous Semantics, where we would find huge amount of knowledge information, but also find most of them transitive along with user context. Therefore, in order for an agent to meet user's real-time query it is required to efficiently retrieve timely and useful piece of the knowledge from the Ubiquitous Semantics. Thus, this paper shows a knowledge-filtering agent, which quickly responds the query by dynamic classification of the necessary information along with the user context changing in the real world. Further, to evaluate our approach we validate the performance of an application: Recipe Recommendation Agent.

## 1 Introduction

Semantic Web[1] has gained attention for recent years. As the popularity of Semantic Web, it is gets for an agent to gather enormous knowledge from Semantic Web. Also, Ubiquitous Computing is expected to become much popular. In Ubiquitous Computing world, people can use computers and networks anywhere-anytime and detect everything with RFIDs.

In near future, data and services would be annotated even in the ubiquitous network, and connected to the web of the semantics. We call it Ubiquitous Semantics, which is an extension of the current Semantic Web. Ubiquitous Semantics is different from Semantic Web in the following points.

1. The agent can retrieve huge amount of knowledge from not only the networks but also people, object and places in the ubiquitous environment. However, most of them are *transitive*, which is described in the next section.
2. In the ubiquitous environment, it is necessary that the agent detects user context and responds quickly in order to support the user's behavior in the real world.

In short, the agent can get huge amount of knowledge from Ubiquitous Semantics, but it is difficult to meet the user's real-time query. Therefore, it is required to retrieve timely and useful piece of the knowledge from the Ubiquitous Semantics according to the user context.

Thus, this paper proposes a knowledge-filtering agent, which quickly responds the query by dynamic classification of the useful information along with the user context changing in the real world. Here, the knowledge is metadata annotated to somethings, which is represented in a triple form including facts, rules, and ontologies.

The rest of this paper is organized as follows: section 2 describes *transitivity*. Section 3 proposes the knowledge-filtering agent based on *transitivity*. In section 4, we overview the architecture of our recipe recommendation agent for evaluation, and validate the performance of the application in section 5. Then, in section 6, we discuss related works, and section 7 concludes this paper.

## 2 Transitivity of Knowledge

The knowledge of the current Semantic Web is sort of static such as web pages. However, in the ubiquitous environment, it is necessary to consider the knowledge changing along with the user's real-time context. In other words, the knowledge in Ubiquitous Semantics must be filtered along with the user's time, place, and so on. We call it *transitive* knowledge.

Therefore, we propose a method to classify the knowledge based on the transitivity and to select a certain size of useful knowledge. This will enable the agent to reason on it efficiently and quickly. In order to classify knowledge based on the transitivity, we define the following four factors of transitivity.

First factor is Time. In the real world, there is much knowledge depending on time. Therefore it is important to select useful knowledge based on the time.

Second factor is Place. In the ubiquitous environment, the user would mainly need to know the knowledge related to the present time and place.

Third factor is Occasion. According to the user's current context, it is different whether the user wants to have a response quickly or not. If the user doesn't have so much time, the agent should inference for short time period and respond quickly. Thus, the user's occasion is an important factor to detect the transitivity of Ubiquitous Semantics.

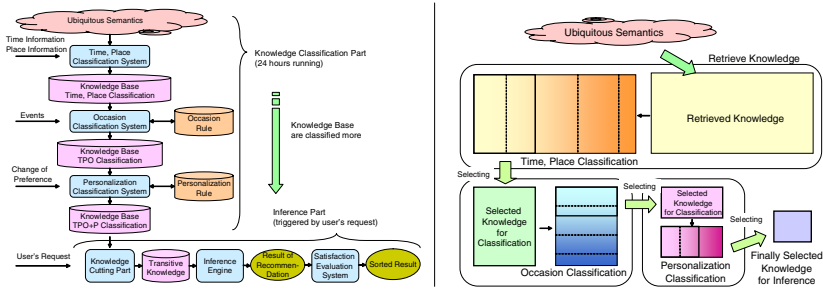
Fourth factor is Personalization. User's preference is also an important factor for selecting. Therefore, the agent should consider the user's preference.

Here, we take the initials of the four names, and call it TPO+P. In the next section, we describe an inference agent who classifies the transitive knowledge based on TPO+P.

## 3 Knowledge-Filtering Agent

The agent needs to select useful knowledge in considering transitive knowledge mentioned in section 2 in order for the agent to respond quickly in Ubiquitous





**Fig. 1.** Architecture of the knowledge-filtering agent and knowledge filtering

Semantics. Figure 1 shows the architecture of the agent. This agent is mainly composed of Knowledge Classification part and Inference part.

The first part including knowledge base vertically connected from top to down is Knowledge Classification part. This part classifies transitive knowledge. The transitive knowledge are classified in three steps based on Time-Place, Occasion and Personalization. Here, we applied the strategy which is to process the simple classification first to make the size of knowledge passed to the next more complex classification smaller. In addition, each classification is processed independently. Therefore, it is possible to re-classify transitive knowledge quickly in case of the change of user's context (figure 1). Knowledge Classification part is always running, then receives input information of user's position, event, preference and so on. We describe each step of classification process and the example in the next section.

The second part including an inference engine horizontally from left to right is Inference part to provide decision support information for users. This part is executed on the user's demand. Knowledge is already retrieved and classified by Knowledge Classification part. Then, the inference part just selects the useful part of knowledge to pass it to inference engine for decision making support. Finally, the agent calculates the satisfaction ratio from the results of inference, and outputs the sorted results with the satisfaction ratio.

## 4 Recipe Recommendation Agent

We have developed recipes recommendation agent for evaluation. The agent recommends a recipe, for example, for homemakers to prepare dinner in consideration of sale information and children's preference and so on. Here, we assume the ubiquitous environment as follows. The information of user's position and what merchandise in user's hand is acquired by using GPS and a RFID reader in the portable device. Also, the agent acquires the necessary knowledge from Ubiquitous Semantics in cooperation with information appliances at home and makes the recommendation. Finally, the portable device displays the recommended recipes.

#### 4.1 Overview of Recipe Recommendation Agent

In the followings, we show the process of classification.

##### 1. Time-Place Classification

Firstly, the agent detects user's position and retrieves knowledge of shops around the user and their sale information and so on, and classifies them based on time and place .

The agent retrieves not only knowledge of business hours and regular holiday of the shop, but also all knowledge that depends on time such as time-sale, then classifies them.

Knowledge Classification part is always running. Thus, the knowledge near the user such as local weather information is updated any time.

The information of shops such as opening hours and position, etc. are assumed to be represented in RDF. In addition, we defined an ontology for shop description (e.g. opening hours, shop holiday, service time and so on). This ontology is defined with DAML-Time ontology[2].

##### 2. Occasion Classification

For example, consider that the user picks up a food stuff in a shop. In this case, it is thought that the user is interested in that food stuff. Thus, the agent should recommend some recipes using it. So the agent retrieves the name of the food stuff from the attached RFID or QR Code, and selects recipe, and recommends some of them. In Occasion classification process, Jess ( Java Expert System Shell ) [3] is used as an inference engine. Therefore, Occasion rules are represented in S-expression like Lisp.

Occasion rules are divided into Common Rules and Condition Rules (table 1). Common rules describe some typical situations and are prepared by the system manager. On the other hand, Occasion Rules describe the situations depends on the user. Therefore, we will develop a tool to select and customize the Occasion Rules in the future.

##### 3. Personalization Classification

Finally, the agent classifies the useful knowledge based on the user's preference, and calculates a satisfaction ratio for recommendation. The user's

Table 1. Occasion rules

Rule Type	Description	Rule
Common Rules	If an user is in a shop, then Agent cut the knowledge small size ( because the user would want to be recommended quickly ).	(defrule (user-in-shop) => (cut-knowledge-small))
	If an user is in a shop, and the shop will be closed soon, then Agent re-check the prices ( because it will be saved ).	(defrule (user-in-shop) (closing-time-soon ?shop) => (check-price ?shop))
	If an user has a car, and is at home, then the user can go shopping by car.	(defrule (user-have-car) (user-in-home) => (use-car-enable))
Condition Rules	If an user is in a shop, and picks up merchandise, then the Agent recommend recipes using it.	(defrule (user-in-shop) (event-have ?item) => (recommend-recipe-use ?item))
	If an user is near a station, and it is the time of going home, then the Agent recommends to buy at the shops in his way home.	(defrule (user-near-station) (time-evening) => (area-shop-station-home))
	If an user is reading a shop handbill, then the Agent recommends to buy at the shop.	(defrule (event-read-handbill ?shop) => (area-shop ?shop))

preference is complex, then it is written as rules. Personalization classification is most time-consumption, so it is executed at the end.

For example, if the user is on a diet, the agent should retrieve knowledge of nutrition information in order to calculate Calorie.

Personalization rules in which the user's preferences are described are specified with URL, and downloaded from the network.

The user's preferences are gathered by questionnaires in advance, then converted to the Personalization rules. We will also develop a tool as well as the Occasion rules in the future, so that the users can describe the Personalization rules by themselves.

## 4.2 Motivation of Recipe Recommendation Agent

The recipe recommendation is one of the best applications for the evaluation as follows.

First, there is much of transitive knowledge in the recipe recommendation. For example, sale information on each day and time service discount information are transitive knowledge depends on time information. In addition, user's preference depends the physical condition on everyday. Therefore, the preference is also transitive knowledge.

Second, lots of sites show several recipes on the web, and various terms are used. Thus, it is necessary to use the ontology in recipe recommendation agent. For example, "Potato" is necessary as a food stuff for a menu, and a certain merchandise is labeled as "White Potato", then the agent should recognize that "White Potato" is one of "Potato" and can be used as the food stuff of the menu. A food stuff ontology is described in Web Ontology Language (OWL) [4].

## 5 Implementation

The mobile device of the recipe recommendation agent is assumed as an advanced cellular phone (Smart Phone). However, the evaluation system was implemented in Pocket PC due to the problem of the development environment (figure 2).



Fig. 2. Mobile device for evaluation

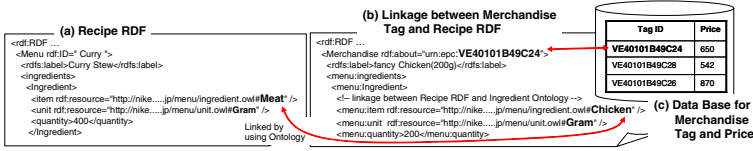


Fig. 3. Knowledge on merchandises and recipes

We use HP iPAQ h2210 in which RFID Reader and GPS Reader are installed. PDA OS is Pocket PC 2003.

We developed the recipe recommendation agent with Java, and installed it to PC instead of a home server. We used Jena 2 Semantic Web Framework [5] to process OWL ontology, and use Jess[3], which is a forward chaining inference engine in Java. Thus, the user’s preferences are written in Jess rules. Web services are provided in some of servers by using Axis[6], and the mobile device communicates with the servers via the agent.

Further, we prepared the knowledge which links the merchandise tag with the recipe written in RDFs as shown in figure 3. Information on tag IDs such as prices are stored in merchandise DB, and the knowledge also links the merchandise tag ID to the food ontology.

## 6 Evaluation

In this section, we evaluate the knowledge-filtering agent. The evaluation was done on the response time and the accuracy.

The knowledge used in the experiment is the real data which is published on a food company [7]. We converted it into RDF and used it as Ubiquitous Semantics. Also, we converted part of the thesaurus which is made by [8] into OWL, and used it as the food ontology.

### 6.1 Response Time

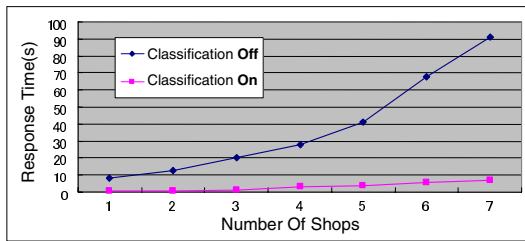
Table 2 shows the result of the response time comparing classifications and non classification. The results are the averages over 3 times sending the same query. We had an experiment in the occasion that the user is in a shop and thinks of today’s dinner then picks up a merchandise at 2 shops, 93 merchandise in each shop, and 40 recipes. The user sends the request to prefer the lowest price.

The agent classifies the knowledge based on shops by “Time-Place classification”, and classifies it based on the merchandise which the user picked up by “Occasion classification”. Moreover, by “Personalization classification” the agent classifies the knowledge based on preference for lower price recipes.

Table 2 shows the response time improvement. It is caused by classifying the knowledge based on TPO+P and selecting suitable knowledge of recipe according to the users context. This result shows that the response time is getting faster by increasing the classification factors.

**Table 2.** Response time with classification factors (ms)

	No Classification	TP	TPO	TPO+P
Response Time	11550	6940	4137	291
Initialization	781	0	0	0
Non-Transitive Knowledge	471	0	0	0
TP Classification	0	317	0	0
TPO Classification	0	0	531	0
Pre-Inference Process	2033	1088	154	0
Price Calculation	7968	5261	3211	0
TPO+P Classification	0	0	0	50
Inference	297	274	241	241

**Fig. 4.** Response time with knowledge size

By looking at the internal processing time, it is found that the calculation on the total price of the recipe takes so much time. The agent infers with the food ontology like section 4.2 by using Jena and calculates the price. Therefore, as the knowledge of the merchandise and the recipes increase, their combinations increase and the processing time grows. However, as the classification factors of knowledge information increase, the combinations become smaller, and the processing time is getting faster.

In addition, we had an experiment on the response time when the size of knowledge is changed. Figure 4 shows the result of classification. When not classifying it, the combinations of the merchandise of the shop and the food stuff of recipes increase explosively. Thus, the response time is getting worse rapidly. In contrast, the response time is almost stable when classifying it.

Further, to confirm whether the classification order TPO+P is appropriate, we shuffled the order. Table 3 shows the results of the time for each order. It is confirmed by this result that the order of “Time-Place”, “Occasion”, and “Personalization” is the fastest, and appropriate as the classification order.

Finally, we had an experiment on a processing time for re-classification with 100 recipes, 200 merchandises, and 4 shops. Figure 4 shows the result. When the re-classification is done at “Time-Place classification”, it is necessary to do re-classification at “Occasion classification” and “Personalize classification” that are below it. That is, it costs the longest time to re-classify the knowledge at “Time-Place classification”. The result shows it takes about 6 seconds to do

**Table 3.** Reclassification time (ms) with TPO+P order

	TimePlace	Occasion	Personalization	Total
TP,O,P	497	251	448	1196
TP,P,O	538	808	855	2201
O,TP,P	631	1519	581	2731
O,P,TP	644	1519	12939	15102
P,TP,O	2507	240	39587	42334
P,O,TP	648	1385	40899	42932

**Table 4.** Reclassification time (ms)

Change level	Reclassification Time
Time-Place changed	6023
Occasion changed	5488
Personalization changed	5021

re-classification at “Time-Place classification”. However, considering with the current PC spec; Pentium M 1.5GHz, the agent would be able to follow enough the user’s real movement.

As the result, the agent can decrease the number of combinations, reduce the size of knowledge for inference, and improve the response time. Also, as the size of ubiquitous semantics increases, the effectiveness of the agent will become higher.

## 6.2 Accuracy of Recommendation

This section shows that the accuracy does not get worse by cutting out the transitive knowledge. The verification method is as follows. First of all, each tester recommends 20 recipes that he/she wants to eat in some occasions from 100 recipes. Then, the recipe agent recommends 20 recipes. Finally, we examine how many recipes which the agent recommended are matched to his / her recommendations, and calculate the precision ratio of the recommendation.

The occasions are the followings.

**Occasion A.** In a shop at 3:00 PM, selecting a food stuff for today’s dinner.

At that time, the user picks up a savory carrot.

**Occasion B.** Around the train station at 10:00 PM. The user buys a food stuff at a convenience store.

**Occasion C.** At home at 3:00 PM, thinking of the menu of today’s dinner.

Here, the user’s preference is “cooking time is shorter”, “low calorie”, “dislikes fishes”. We had an experiment with 230 merchandise, 231 food ontology, and 7 testers. Table 5 shows the result of the average. (No-classification experiment was not able to be done, because the system rised memory shortage error.)

**Table 5.** Precision ratio (%)

	TP	TPO	TPO+P
Occasion A	43.3	83.3	83.3
Occasion B	35.0	46.7	51.7
Occasion C	30.0	40.0	43.3

The result shows that the precision ratio improves as the classification factor increases. In each classification, obviously unsuitable recipes are cut out, so the precision ratio of the recipe has improved.

In summary, it was confirmed that the agent is able to respond quickly keeping the accuracy by classifying transitive knowledge based on TPO+P.

### 6.3 Applicability of Other Applications

We have concluded the knowledge filtering agent is effective by evaluating the recipe recommendation system only. However, the knowledge filtering agent is applicable to other applications. For example, man navigation system which changes the destination along with user's preference is one of target applications. This system has too many possible destination for users goal. Therefore, it is difficult to consider the whole knowledge. Furthermore, it is necessary to recommend quickly in the case of changing the user's plan. Also, there are many kinds of transitive knowledge, such as vacant seat in the theater and so on. For the above reasons, the knowledge filtering agent would be applicable for other systems.

## 7 Related Works

Several studies have been made on context awareness in ubiquitous environment. [9] aims at providing Web services that fit to ubiquitous computing and proposes an architecture with middle agents who determine the best matched services and location-ontology for ubiquitous computing. However, it doesn't classify the knowledge information from huge amount of transitive knowledge.

[10] proposes a system which infers user's context from the knowledge in Semantic Web and information from sensors, and provides appropriate information to the user. However, it don't classify massive transitive knowledge and not consider the performance.

In addition, several methods of acquiring knowledge to respond quickly are proposed. [11, 12] propose an agent who acquires the knowledge on the Web using caching and planning technology. However, they don't deal with transitive knowledge in ubiquitous environment. Our research aims to respond quickly by classifying transitive knowledge information.

Furthermore, several studies have been made on recipe recommendation. [13] proposes a system to recommend new recipes from some basic recipes by using Case-Based Reasoning and propose a substitute food stuff by using a food ontol-

ogy. However, it doesn't consider the transitive knowledge. If the size of knowledge is large, then it would become necessary to select the useful knowledge.

## 8 Conclusion

We defined four factors that characterize the transitive knowledge as TPO+P, and proposed the method of efficiently selecting the useful knowledge part from the huge amount of knowledge in ubiquitous environment.

Then, we developed the recipe recommendation agent, and evaluated the response time and the accuracy.

## References

1. Tim Berners-Lee, James Hendler, and Ora Lassila. The Semantic Web. *Scientific American*, May 2001.
2. Harry Chen, Filip Perich, Tim Finin, and Anupam Joshi. SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications. In *MobiQuitous 2004*, August 2004.
3. Jess (Java Expert System Shell). Sandia National Laboratories. <http://herzberg.ca.sandia.gov/jess/>.
4. Deborah L. McGuinness and Frank. van Harmelen. OWL Web Ontology Language Overview, December 2003. <http://www.w3.org/TR/owl-features>.
5. Jeremy J. Carroll, Ian Dickinson, Chris Dollin, Dave Reynolds, Andy Seaborne, and Kevin Wilkinson. Jena: Implementing the semantic web recommendations. Technical Report HPL-2003-146, HP Lab, 2003.
6. Apache AXIS. Apache Web Services Project. <http://ws.apache.org/axis/>.
7. Ajinomoto, Co., Inc. Recipe *DAIHYAKKA*. <http://www.ajinomoto.co.jp/recipe/>.
8. Institute of Language Engineering. Thesaurus. Japan, <http://www.gengokk.co.jp/thesaurus/>.
9. Akio Sashima, Koichi Kurumatani, and Noriaki Izumi. Location-mediated service coordination in ubiquitous computing. In *the Third International Workshop on Ontologies in Agent Systems(OAS-03)*, pages 39–46. AAMAS2003, 2003.
10. Harry Chen, Tim Finin, Anupam Joshi, Filip Perich, Dipanjan Chakraborty, and Lalana Kagal. Intelligent Agents Meet the Semantic Web in Smart Spaces. *IEEE Internet Computing*, 8(6):69–79, November 2004.
11. Victor Lesser, Bryan Horling, Frank Klassner, Anita Raja, Thomas Wagner, and Shelley XQ. Zhang. BIG: An Agent for Resource-Bounded Information Gathering and Decision Making. *Artificial Intelligence*, 118(1–2):197–244, May 2000.
12. Victor Lesser, Bryan Horling, Frank Klassner, Anita Raja, Thomas Wagner, and Shelley XQ. Zhang. BIG: A Resource-Bounded Information Gathering and Decision Support Agent. Technical Report 1998-52, Multi-Agent Systems Laboratory Computer Science Department University of Massachusetts, January 1999.
13. Kristian J. Hammond. CHEF: A Model of Case-Based Planning. *AAAI*, pages 267–271, 1986.



# Application-Driven Customization of an Embedded Java Virtual Machine

Alexandre Courbot<sup>1</sup>, Gilles Grimaud<sup>1</sup>,  
Jean-Jacques Vandewalle<sup>2</sup>, and David Simplot-Ryl<sup>1</sup>

<sup>1</sup> IRCICA/LIFL, Univ. Lille 1, France, INRIA futurs, POPS research group  
{Alexandre.Courbot, Gilles.Grimaud, David.Simplot}@lifl.fr

<sup>2</sup> Gemplus Systems Research Labs, La Ciotat, France  
Jean-Jacques.Vandewalle@research.gemplus.com

**Abstract.** Java for embedded devices is today synonym of “embeddable pseudo-Java”. Embedded flavors of Java introduce incompatibilities against the standard edition and break its portability rule. In this paper, we introduce a way to embed applications written for Java 2 Standard Edition. The applications are pre-deployed into a virtual Java execution environment, which is analyzed in order to tailor the embedded Java virtual machine according to their runtime needs. Experiments reveal that this method produces customized virtual machines that are comparable in size to existing embedded Java solutions, while being more flexible and preserving standard Java compatibility.

## 1 Introduction

Many solutions exist as of today for using Java on small and restrained devices [1], like Java 2 Micro Edition (J2ME) and Java Card. To become embeddable, these flavors deviate from standard Java and only offer a subset of its features. Java 2 Standard Edition (J2SE [2]) is the original edition of Java, and as such has the widest applicative spectrum of all the Java implementations. However, its resource requirements limit it to desktop workstations or strong PDAs. Lighter devices have to turn to degraded versions of Java such as J2ME. These Java flavors come with APIs that cover a limited range of the J2SE APIs, and are sometimes incompatible with it. In addition, their virtual machine doesn't cover all the features range of the J2SE specification. A Java derivative is therefore only suitable for a given kind of applications and a given range of devices, and enforces the application programmer to cope with an environment that is not J2SE-compliant. The portability gold rule of Java is thus broken.

Obviously, using Java on restrained devices requires a degradation of the Java environment at some point to make it fit. However, imposing restrictions right from a specification tend to make the environment either suitable for the general use and inefficient for dedicated tasks, or good for one domain and inapplicable to others. It also multiplies the number of incompatible implementations of Java that a developer has to choose from. Our approach is to tailor the most suitable customized Java environment from a standard Java environment on a per-case

basis, according to the applications that are to be run, and the specifics of the target device. As efficient customization of software requires knowledge about its runtime conditions, the customizations take place during an off-board pre-deployment phase of the system, called *romization*.

We identify two kinds of customizations that are applicable during romization. The first one, automatic reduction and specialization of the J2SE APIs to get light and efficient custom-build APIs, has been studied in previous work. In the present paper, we are interested in the specialization of the Java virtual machine that is embedded into the target device. We propose and evaluate a method for determining and removing the virtual machine features that are not necessary to the embedded applications. This approach has the advantage to retain J2SE compatibility for the programmer, and to provide an adequately-tailored virtual machine to the applications.

The remainder of this paper is organized as follows. In section 2, we make an overview of Java on embedded devices, introduce the romization concept, and summarize our previous work on it. Then, section 3 explains how deployment-time analysis of the system can be useful to customize the embedded virtual machine. Section 4 experimentally measures the memory gained by removing unused virtual machine features, and we conclude on our approach in section 5.

## 2 Overview of Java on Embedded Devices

In this section, we overview some existing solutions for using Java on small and restrained devices. Then, we present the romization process, its advantages for embedded Java systems, and summarize our previous work on it.

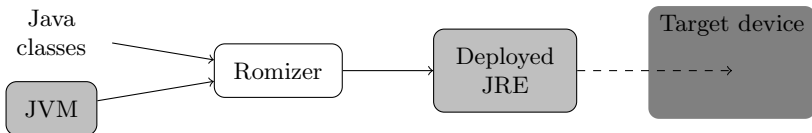
### 2.1 Java on Embedded Devices

Java offers features like compact program bytecode and safe execution that make embedded Java a hot topic. As of today, many embedded Java environments are available. Java 2 Micro Edition (J2ME [3]) specifies a Java-like virtual machine specification and APIs, and is derived into two *configurations*. The *Connected Device Configuration* (CDC) is designed for network appliances, while the *Connected Limited Device Configuration* (CLDC) is targeted at small and mobile networked devices, like mobile phones. Both CDC and CLDC come with a small subset of the J2SE APIs and bring new, incompatible APIs. Moreover, CLDC imposes restrictions on the virtual machine: no support for reflection, objects finalization, floating point numbers, and limited error handling. Java Card [4] is another Java derivative from Sun that targets smartcards. It has more limitations than CLDC, since it also drops support for garbage collection, 32-bits operands, and strings. Java Card also deviates by the firewalling mechanism, and its `.cap` preloaded class format. TinyVM and LeJOS [5, 6] are community projects for enabling Java on the Lego Mindstorm platform. They propose two differently sized and featured implementations, with additional non-standard APIs and limitations on the virtual machine.

Java’s promise is “*compile once, run everywhere*”. But as we can see, all the embedded solutions considered here are incompatible with J2SE and violate this rule. Moreover, they offer a rather static virtual machine configuration, which features may not all be exploited by the embedded applications, thus wasting silicon and questioning the relevancy of writing small applications in Java. To address these issues, we propose not to adapt the applications to a specific Java environment, but on the contrary to tailor a standard J2SE environment according to its applications and targeted device. Such a tailored Java system becomes embeddable and provides the right subset of runtime features needed by the applications. Our approach, which customizes the J2SE APIs as well as the embedded virtual machine, relies on the romization process.

## 2.2 The Romization Process

Romization is the process by which a Java system is pre-deployed by a deployment host, for a target device. In this particular form of deployment, the device that runs the system is not the device that deployed it. Romization differs from distributed deployment methods like Java Card `.cap` format or JEFF [7], which are pre-loaded alternatives to the `.class` format. The romizer deploys the Java Runtime Environment within a virtual execution environment, and then dumps a memory image of it. This memory image containing the deployed system is then copied to the target device where it will continue its execution. Romization can therefore be characterized as an “in-vitro” form of deployment (figure 1).



**Fig. 1.** The romization process

Romization brings two interesting properties for restrained devices willing to run Java. First, the device does not need to support the cost of deployment. This point is important because Java class loading is too costly a process for many small devices. Second, the output of the romizer serves as the initial state for the device (the state it is in when powered on). Since this state comprehends the deployed Java virtual machine, the device is ready to use it immediately, which reduces startup times. These points make romization a very common practice, not to say a mandatory step, in the embedded Java world.

## 2.3 Previous Work on Romization

As of today, romization is primarily used to pre-load Java classes and provide a service quite similar to distributed class formats. In previous work [8], we have

overridden this classical usage of romization and shown the benefits of going further in the system deployment during romization: the romizer can perform very aggressive customizations on the system if the latter is deployed far enough. In particular, call graph analyses [9] on the threads allow unused parts of the APIs to be removed using library extraction techniques [10, 11], and the remainder to be specialized for runtime usage. This results in APIs that are custom-tailored on a per-case basis for the system, and have low memory footprints.

This previous work did only cover the specialization of the deployed applications and Java APIs. In this paper, we take advantage of the advanced deployment state of the system to customize the embedded Java virtual machine.

### 3 Customization of the Java Virtual Machine

We have seen in the previous sections that many features of the J2SE virtual machine are not supported by restrained devices. We are however interested in developping our applications using the standard J2SE, and degrading it according to the applications needs and the capabilities of the target device. This section evaluates how the necessary runtime features of a Java program can be figured out, while the next one gives experimental results on this approach.

The purpose of the Java virtual machine is to execute Java programs: i.e, to provide an implementation for every bytecode used by a program, in such a way that its semantic is conform to the Java specification. If a bytecode is not used by the virtual machine, support for it can safely be dropped.

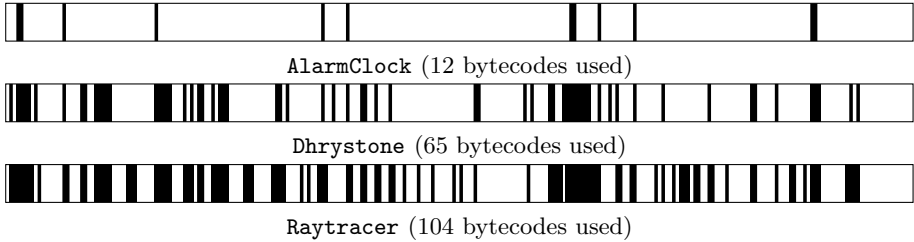
#### 3.1 Unused Bytecodes Support Removal

The full Java instruction set covers a large spectrum of operations: integer and floating point arithmetic and logic for 32 and 64 bits operands, objects allocation, methods invocation, threads synchronization, and so on. But few Java programs use all the bytecodes – this is especially true for small programs. For instance, many embedded applications have no use for floating point arithmetic. Critical applications, if deployed far enough within the romizer, often never allocate memory.

Figure 2 shows the bytecodes usage spectrum of various benchmark programs, as stated by the call graph analysis done in our romizer. **AlarmClock** is a simple alarm program that waits for a given time to be reached. **Dhrystone** is the well-known integer operations performance benchmark, and **Raytracer** is a multi-threaded image rendering benchmark from the SPEC JVM98 suite [12].

It is striking that every benchmark is far from using all the bytecodes of the Java instruction set. A very small application like **AlarmClock**, which scope is limited to integer arithmetic, has no use for the majority of them. **Dhrystone** uses strings and is already a more complete program, but there are still more white sections than black ones on its spectrum. **Raytracer** heavily uses floating point arithmetic in addition to integers, as well as threads synchronization and memory allocation. However, it still uses less than half of the bytecodes set.

Once the set of useful bytecodes is determined, support for unused ones can be removed from the bytecode interpreter. This may give the opportunity to



**Fig. 2.** Bytecodes usage spectrum for different programs. The horizontal axis parses the whole Java instruction set. Bytecodes present in the program call graph rise a black bar, whereas a white gap indicates an unused bytecode.

remove some services provided by the virtual machine. For instance, the `new` bytecode is responsible for allocating memory for an object of a given class. To do so, it uses a virtual machine function that allocates memory on the heap. If there is not enough memory available, this function triggers a garbage collection to recover memory. This mechanism is reproduced for other memory allocation bytecodes, like `newarray` or `anewarray`. If the `new` bytecode is never to be met by the virtual machine at runtime, the object allocation function of the virtual machine can be dropped. If none of the memory allocation bytecodes is present in the program code, then not only can their corresponding allocation functions be dropped, but also the garbage collector since it is never going to be useful to recover memory: the gain of removing all the memory allocation bytecodes is greater than the cumulated gain of removing each of them individually.

All the bytecodes are not equally interesting to remove. Memory allocation bytecodes are great candidates, because they rely on heavy mechanisms. On the contrary, removing an arithmetic bytecode leads to a poor gain. We noticed that two-thirds of the memory footprint of the virtual machine serves for implementing one-tenth of the bytecodes: those responsible for memory allocation, threads synchronization, exception throwing, and method invocation.

Removing support for bytecodes in the virtual machine is a good way to eliminate some of its useless features. However, all the virtual machine features are not exclusively dependent on the presence of some bytecodes. For instance, threads switching may be triggered by a bytecode (`monitorenter` or `monitorexit`), but also by other events (a thread used its time slot, a native method put the current thread in sleeping state, ...). Such features require a deeper analysis of the system in order to be decided useful or not.

### 3.2 Analysis of the Deployed System

Some virtual machine features, such as threads management, would always be present in the virtual machine no matter the bytecodes included. The reason is that these mechanisms are called by the virtual machine itself: for instance, when a time slice is elapsed, the virtual machine requests a thread switching. This is unfortunate because threads management is responsible for a non-neglectable

part of the virtual machine memory footprint, and some Java systems have no use for threads (for instance, Java Card). In particular, systems that never have more than one Java thread simultaneously still perform in accordance with the Java specification if they don't include multithreading.

In a virtual machine developed with configurability in mind, threads management can easily be disabled through compile-time definitions. In such a configuration, the bytecode interpreter executes the current thread without switching and ends with it. It is possible for the program analyzer of the romizer to detect in which case this configuration is possible. The virtual machine can be purged of threads management if it fulfills the following conditions:

- There is only one active thread into the system at the time of analysis,
- The program analysis reveals that the method `Thread.start()` is never called,
- No additional code is loaded from the outside.

In such a case, it is assured that no more than one thread will ever run, and the threading mechanisms of the virtual machine can safely be discarded by a compilation flag.

The system analysis might enter in conflict with the bytecodes support removal in some cases. For instance, consider a system which fulfills the conditions to be mono-threaded. However, its execution path meets the bytecode `monitorenter` at some point (for instance, by calling a synchronized method). The implementation of `monitorenter` triggers a thread switching if the current thread doesn't own the monitor, thus including the thread switching functions into the virtual machine. To override this problem, all the instances of `monitorenter` and `monitorexit` in the Java code can be eliminated during romization, which also has the beneficial side effect of reducing the code size.

Removing threads management when it is useless is just an example amongst others, although it is probably the one that offers the most significant memory gains. Similar analyzes can be performed for other customizations, like disabling support for exceptions.

## 4 Experimental Evaluation

The previous section explained how to detect and remove features of the virtual machine unneeded for a given Java program. In this section, we evaluate the effective memory footprint gained by this tailoring.

### 4.1 Methodology

All our measurements have been performed on the Java In The Small (JITS [13]) Java-OS. JITS comprises J2SE-compliant APIs and virtual machine, and a romization architecture that allows to execute the system off-board and to perform analyzes on it. The binaries are obtained by romizing the benchmark programs mentioned in section 3, and by compiling the tailored JITS virtual machine

using GCC 3.4.3 with optimization level 2, for the x86 architecture. The linker is then asked to eliminate dead code.

The JITs virtual machine is made of several compilation units. The bytecode interpreter engine, when including support for all bytecodes, is 15350 bytes big. The interpreter loop itself is 11670 bytes, the rest are peripheral features like method frame creation or exception throwing. The full threads management mechanisms take 6967 bytes, and the complete memory manager 10915 bytes, of which 7036 are for the garbage collector. A non-customized JITs virtual machine therefore has a memory footprint 33232 bytes, to which one must add the target-specific code, native methods, and Java classes that are needed for the virtual machine to function. Indeed, many core features of JITs, like the class loader, are written in Java. We are not including these features in our measurements because they are covered by the customization of the Java classes that has been addressed in previous work. In this paper, we are just interested in tailoring the natively-written part of the runtime environment.

## 4.2 Results

Table 1 shows the sizes obtained for virtual machines capable of running our benchmark programs.

**AlarmClock** uses 12 bytecodes, and its engine is reduced to 2895 bytes. This program never allocates memory, which makes the memory manager unnecessary, and the threads management operations can also be highly reduced since the program never creates new threads. Moreover, the set of bytecodes used is quite “ideal”, with only low-cost bytecodes (stack manipulation and integer arithmetic). This explains the very low footprint of this virtual machine.

**Dhrystone** uses 65 bytecodes, for an engine size of 6992 bytes, and is also mono-threaded. One question can be raised about why the memory manager is not included in the binary, since this benchmark allocates arrays in its source code. The answer is, because all the allocations have already been performed within the romizer. **Dhrystone** allocates memory at two points of its execution: during the initialization of the classes (for initializing static fields), and at the very beginning of the benchmark where it allocates one-sized integer arrays (which are a trick to simulate passing integers by address). These memory allocations are not performed at runtime because the romizer dumped the state

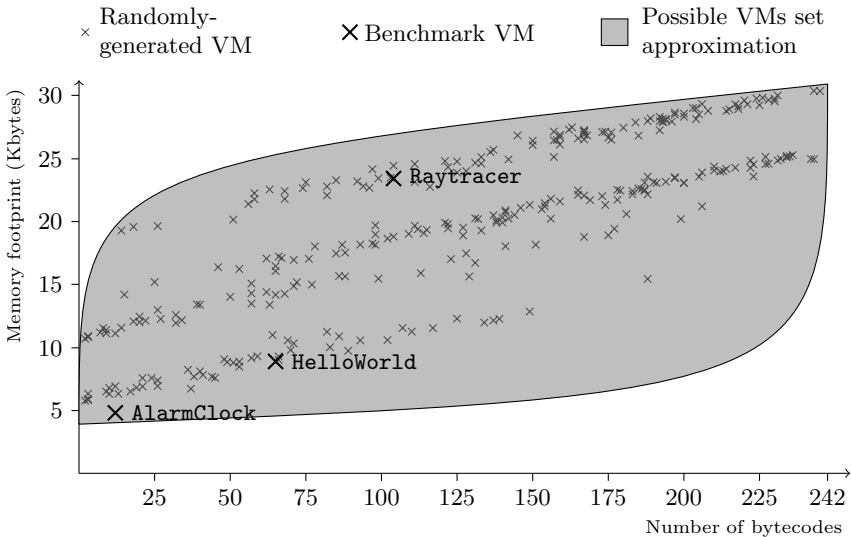
**Table 1.** Size (in bytes) of the obtained virtual machines for the different benchmark programs

Benchmark	Reference	AlarmClock	Dhrystone	Raytracer
Number of bytecodes used	242	12	65	104
Engine size	15350	2895	6992	8576
Memory manager size	10915	0	0	7986
Threads management size	6967	1908	1908	6854
<b>Total size</b>	<b>33232</b>	<b>4803</b>	<b>8900</b>	<b>23416</b>

of the system *after* their execution. It doesn't change the runtime semantic of the program because at this point the benchmark algorithm has not yet started. Therefore, these memory allocations can be considered initialization work that is safe to be performed off-board. This is a typical example of the advantage of bringing the system to an advanced state of deployment within the romizer: if our romization architecture were only capable of pre-loading the classes, none of these initializations would have been performed. Our virtual machine would then have suffered a penalty of several kilobytes for the memory manager; not to mention the footprint of the class initialization mechanism, and the increased startup time of the system.

We should also mention that the customization of the Java APIs done during romization is essential for efficiently removing bytecodes. For instance, the `String.charAt` method used by `Dhrystone` allocates and throws an exception if the given index isn't within the range of the string. But since `Dhrystone` always calls this method with well known values and on strings we statically know the size of, the romizer can infer that the exception is never thrown, and improve the code of the method accordingly. Without this APIs customization pass, the exception throwing would be a plausible program path and the `new` bytecode marked as used, requiring a part of the memory allocation module and the whole garbage collector to be included.

Our last benchmark, `Raytracer`, requires 104 bytecodes for an engine size of 8576 bytes. There is no way to completely drop the memory manager since it allocates objects at runtime. Being multithreaded, it also requires almost all the threading mechanisms. Its virtual machine size is therefore of 23416 bytes,



**Fig. 3.** Memory footprint against number of bytecodes supported, for 300 randomly-generated virtual machines. The grey area gives a theoretical approximation of the range of virtual machines that can be generated.



which is only 8 Kbytes less than the fully-featured reference virtual machine. Indeed, **Raytracer** doesn't even use half of the bytecodes set, but within the used bytecodes are a good part of the "critical bytecodes" that require the heaviest features of the virtual machine, notably memory allocation bytecodes.

To complete these experiments, we have generated 300 virtual machines, each one supporting a random number of randomly-chosen bytecodes. Whether the virtual machine is mono or multi-threaded is also determined randomly. These virtual machines are not designed for a particular application, but give an overview of the possible memory footprints for a customized virtual machine.

Figure 3 shows the memory footprints obtained for virtual machines supporting a given number of bytecodes and for our benchmark programs. The grayed area is a theoretical range of the possible virtual machines, based on the individual cost of the bytecodes: the upper curve follows the worst possible case (costly bytecodes included first), while the lower one shows the best case (cheapest bytecodes first). As we can see, the memory footprint varies a lot for virtual machines with the same number of bytecodes. We also notice that the dots tend to group into lines that grow linearly, each line corresponding to the inclusion of a "critical" virtual machine feature: namely, the memory and threads managers. After 150 bytecodes, chances are very low not to include at least one memory allocation bytecode, and the dots form two parallel lines: the lower line for mono-threaded virtual machines, the upper line for multi-threaded ones.

We can compare these results with existing embeddable virtual machines. A standard KVM supporting the CDC configuration is about 40 Kbytes of code when compiled for x86. Recent work on the Squawk virtual machine [14], which aims at providing an efficient CLDC-compliant virtual machine for next-generation smart cards, resulted in a virtual machine memory footprint of 26 Kbytes. Our results obtained by customizing a J2SE virtual machine are therefore quite comparable with these more static solutions. It should be noted, when comparing these sizes with our measures, that the KVM and Squawk footprints comprehend system parts like the class loader which are not included in our virtual machines. This is because the JITs class loader is implemented in Java and is not a direct part of the virtual machine.

## 5 Conclusion

We gave a proposal solution to the problem of embedding Java on embedded and restrained devices. Current solutions consist in statically-degraded Java virtual machines that are incompatible with J2SE. On the contrary, our approach let the programmer use a full-fledged J2SE virtual machine, which is then customized during romization according to the applications it is going to run and the target device that will host it. The "right" virtual machine is thus generated on a per-case basis, which efficiently reduces its memory footprint.

Put together with our previous work of [8], which tailors the J2SE APIs, these results make it possible to use J2SE for programming embedded Java applications, while providing lower memory footprints than traditional solutions.

Since the bytecodes set is chosen according to the romized applications, this solution is particularly suitable for closed systems that do not load code dynamically. Open systems can define a set of “authorized” bytecodes that are to be included into the virtual machine regardless of their usage by the romized applications; this is especially pertinent if this set only comprehend low-cost bytecodes which gain is negligible. For cases where the Java system has already been deployed, the device memory can also be flashed with another, more featured Java virtual machine.

## References

1. D. Mulchandani, “Java for embedded systems,” *Internet Computing, IEEE*, vol. 2, no. 3, pp. 30 – 39, 1998.
2. T. Lindholm and F. Yellin, *Java Virtual Machine Specification*. Addison-Wesley Longman Publishing Co., Inc., 1999.
3. Sun Microsystems, *J2ME Building Blocks for Mobile Devices*, 2000.
4. Z. Chen, *Java Card Technology for Smart Cards: Architecture and Programmer’s Guide*. Addison-Wesley Longman Publishing Co., Inc., 2000.
5. “TinyVM.” <http://tinyvm.sourceforge.net/>.
6. “LeJOS.” <http://lejos.sourceforge.net/>.
7. The J-Consortium, *JEFF Draft Specification*, March 2002.
8. A. Courbot, G. Grimaud, and J.-J. Vandewalle, “Romization: Early deployment and customization of java systems for restrained devices,” Tech. Rep. RR-5629, INRIA Futurs, Lille, France, July 2005.
9. D. Grove, G. DeFouw, J. Dean, and C. Chambers, “Call graph construction in object-oriented languages,” in *OOPSLA ’97: Proceedings of the 12th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications*, (New York, NY, USA), pp. 108–124, ACM Press, 1997.
10. D. Rayside and K. Kontogiannis, “Extracting java library subsets for deployment on embedded systems,” *Sci. Comput. Program.*, vol. 45, no. 2-3, pp. 245–270, 2002.
11. F. Tip, P. F. Sweeney, and C. Laffra, “Extracting library-based java applications,” *Commun. ACM*, vol. 46, no. 8, pp. 35–40, 2003.
12. “SPEC JVM98 benchmarks.” <http://www.spec.org/jvm98>.
13. “Java In The Small.” <http://www.lifl.fr/RD2P/JITS/>.
14. N. Shaylor, D. N. Simon, and W. R. Bush, “A java virtual machine architecture for very small devices,” in *Proceedings of the 2003 ACM SIGPLAN conference on Language, compiler, and tool for embedded systems*, pp. 34–41, ACM Press, 2003.

# A Study on Fast JCVM with New Transaction Mechanism and Caching-Buffer Based on Java Card Objects with a High Locality\*

Min-Sik Jin<sup>1</sup>, Won-Ho Choi, Yoon-Sim Yang, and Min-Soo Jung<sup>2</sup>

Dept of Computer Engineering, Kyungnam University, Masan, Korea  
{comsta6, hoya9499, ysyang}@kyungnam.ac.kr  
msjung@kyungnam.ac.kr

**Abstract.** Java Card is now a mature and accepted standard for smart card and SIM technology. Java Card is distinguished primarily by its independence from hardware platforms and portability and is now the most important open standard. However, the main concern of Java Card is now its low execution speed caused by the hardware limitation. In this paper, we propose how to improve a execution speed of Java Card by reducing the number of EEPROM writing. Our approaches are an object-buffer based on a high locality of Java Card objects, the use of RAM, has a speed more faster 1000 times than EEPROM, as much as possible and new transaction mechanism using RAM.

## 1 Introduction

Java Card technology [1, 2, 3] enables smart cards and other devices with very limited memory to run small applications, called applets, that employ Java technology such as a platform independence and a dynamic downloading(post-issuance). For these reasons, Java Card technology is an accepted standard for smart card and SIM technology [15]. SIM cards are basically used to authenticate the user and to provide encryption keys for digital voice transmission. However, when fitted with Java Card technology, SIM cards can provide transactional services such as remote banking and ticketing, and also service a post-issuance function to manage and install applications in cards after the cards issued [1, 3, 15].

Java Card uses generally RAM and EEPROM. The difference of both memory is that writing operations to EEPROM are typically more than 1,000 times slower than to RAM. In a traditional Java Card, the specific area, transactionbuffer(T\_Buffer), in EEPROM is used to support an atomicity and transaction [1, 3]. It makes the speed of the Java Card more slowly. In addition to the transaction mechanism, a traditional Java Card has a low-level EEPROM writing with a page-buffer. The size of a page-buffer depends on platforms such as ARM, Philips and SAMSUNG [15]. This page-

---

\* This work is supported by Kyungnam University Research Fund, 2005.

<sup>1</sup> Ph.D Student of Kyungnam University.

<sup>2</sup> Professor of Kyungnam University.

buffer is just to write one byte or consecutive bytes less than the size of the page-buffer at a time into EEPROM. However, this page-buffer of Java Card generally is made regardless of a high locality of Java Card Objects [5, 7].

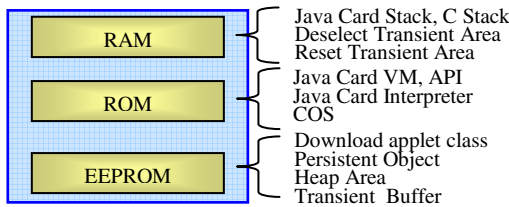
In this paper, we suggest two ideas to improve the speed of Java Card. One is new transaction mechanism in RAM, not EEPROM. Another is new object-buffer based on a high locality of Java Card objects to support a caching and buffering of heap area.

This paper is organized as follows. Section 2 describes the feature of each memory in a typical Java Card, Java Card objects and the method that writes data to EEPROM. Section 3 explains about a transaction and object writing of a traditional Java Card using a lot of EEPROM writing. Section 4 outlines our approach about new transaction mechanism using RAM and new object-buffer based on a high locality of Java Card objects. Section 5 discusses the evaluation between a traditional one and our approach. Finally, we present our conclusions in Section 6.

## 2 The Java Card Environment

### 2.1 Different Types of Memory in Java Card

A typical Java Card system places the JCRE code(virtual machine, API classes) and COS in ROM. RAM is used for temporary storage. The Java Card stack is allocated in RAM. Intermediate results, method parameters, and local variables are put on the stack. persistent data such as post-issuance applet classes, applet instances and longer-lived data are stored in EEPROM [3,5].



**Fig. 1.** The general memory model of Java Card that is consisted of three areas and its contents

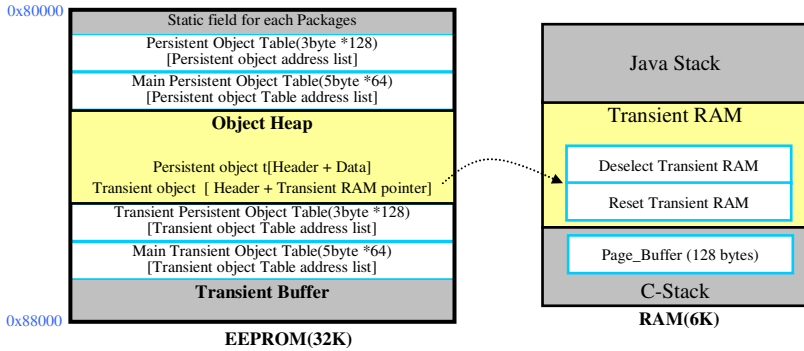
EEPROM provides similar read and write access as RAM does. However, The difference of both memory is that writing operations to EEPROM are typically more than 1,000 times slower than to RAM and the possible number of EEPROM writing over the lifetime of a card is physically limited [4].

**Table 1.** Comparison of memory types used in Smart Card microcontrollers [4]

Type of Memory	Number of write/erase cycles	Writing time per memory access	Typical cell size with 0.8- $\mu\text{m}$ technology
RAM	unlimited	70 ns	1700 $\mu\text{m}^2$
EEPROM	10,000 – 1,000,000	3-10 ms	400 $\mu\text{m}^2$

## 2.2 How to Write Objects in EEPROM in Java Card

In the latest release, Java Card 2.2.1, one EEPROM mainly consists of 3 areas; *static field area*, *heap area* to save many Java Card objects including transient object table(TOT) and persistent object table(POT) and *transactionbuffer(T\_buffer area)*[7].



**Fig. 2.** The inner structure of RAM and EEPROM consisting of several areas. Especially, all objects that are made by Java Card is saved in Heap area with a high locality.

A transaction mechanism [10] using the T\_Buffer area in EEPROM is used to support an atomicity [3]. In a traditional Java Card, to support this transaction, the Java Card temporarily saves old\_data in T\_Buffer in EEPROM until the transaction is complete.

In a point of COS's view lower level than Java Card, smart cards such as Java Card use only one page-buffer in RAM to write data in EEPROM,. The size of the page-buffer depends on platforms such as ARM, Philips and SAMSUNG. In fact, the data is first written into the page-buffer in RAM, when the Java Card writes one byte or consecutive bytes less than the size of the page-buffer into EEPROM. However, the most important point about writing operation using the page-buffer is that the writing time of both 1 byte and 128 consecutive bytes is almost the same.

## 3 A Transaction and Object Writing of a Traditional Java Card

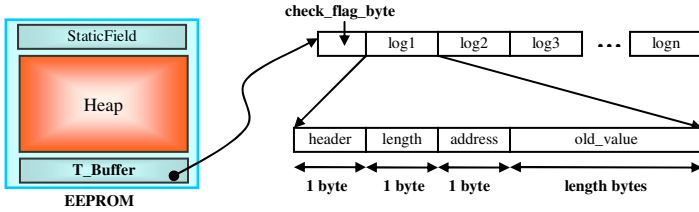
### 3.1 Atomic Operations and Transaction in a Traditional Java Card

A transaction is a set of modifications performed atomically, which means that either all modifications are performed or none are performed. This is particularly for smart cards, because the card reader powers them: when you unexpectedly remove the card from the reader (this is called "tearing"), it's possible that you're interrupting a critical operation that needed to run to completion. This could put the card in an irrecoverable state and make it unusable.

To prevent this, the Java Card platform offers a transaction mechanism. As soon as a transaction is started, the system must keep track of the changes to the persistent environment(EEPROM). The Java Card must save old\_value of EEPROM address

that will be written into a particular area(T\_Buffer) in EEPROM. In other words, If a transactional computation aborts, the Java Card must be able to restore old\_value from the T\_Buffer in EEPROM to its pervious position.

In case of commit, the check\_flag\_byte of the T\_Buffer must just be marked invalid and the transaction is completed. In case of abort, the saved values in the buffer are written back to their former locations when the Java Card is re-inserted to CAD.



**Fig. 3.** The inner structure of T\_buffer has a lot of logs and each log consists of 4 parts; header, length, address and old\_value

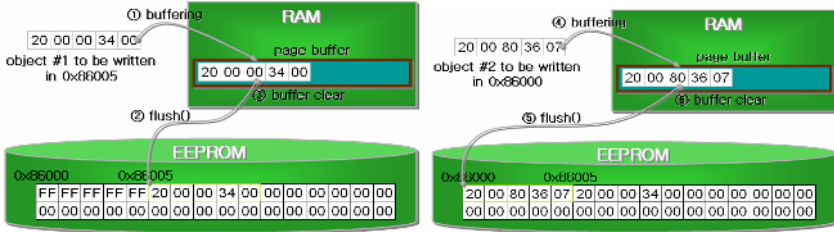
Table 2 below shows the number of EEPROM writing per each area of whole EEPROM. T\_buffer area writing is about 75 to 80 percent of total number. The reason why the writing number of this area is higher than other areas is a transaction mechanism of a traditional Java Card to guarantee an atomicity. In other words, The transaction mechanism protects data corruption against such events as power loss in the middle of a transaction. In a traditional Java Card, this transaction mechanism makes the Java Card more slow and inefficient. In this paper, we suggested new transaction mechanism using RAM, not EEPROM.

**Table 2.** The number of EEPROM writing per each area of whole EEPROM during the downloading and executing of each applet

EMV Applet		Wallet Applet	
EEPROM area	the number of writing	EEPROM area	the number of writing
StaticField	1,681	staticfield	752
Heap	1,659	Heap	1,121
<b>T_buffer</b>	<b>10,121</b>	<b>T_buffer</b>	<b>8,478</b>
<b>Total</b>	<b>13,461</b>	<b>Total</b>	<b>10,351</b>

### 3.2 A Traditional Java Card with One Page Buffer

In a general Java Card environment, one page-buffer in RAM is used to write data into EEPROM. the size of a page-buffer depends mainly on platforms. It is between 128 and 256 bytes. our chip with CalmCore16 MPU from SAMSUNG has 128 bytes page buffer that a Java Card can write up to 128 consecutive bytes to EEPROM at a time. Namely, a Java Card can write between 1 byte and 128 consecutive bytes with this page buffer into EEPROM. For example, If EEPROM addresses of objects that will be written by a Java Card are sequentially 0x86005 and 0x86000, although both addresses are within 128 bytes, Java Card will first writes one object data in 0x86005 through the page-buffer, and then, after the page-buffer is clear, another object data will be written in 0x86000.



**Fig. 4.** how to write objects to EEPROM of the traditional Java Card using an inefficient page-buffer algorithm

Above figure 4 shows the page-buffer algorithm of a traditional Java Card. this page-buffer is just to write consecutive data to EEPROM. It dose not have the function for caching. When an applet is executed on Java Card, if the information such as objects and class data that the applet writes are close to each other, the total number of EEPROM writing would be reduced by adding a caching function to the page-buffer. first of all, to do this, the writing address of objects and data created by Java Card must have a high locality. It causes the number of EEPROM writing to reduce and also makes a hitting rate of caching function more high.

We investigated a general tendency of writing operation in accordance with the EEPROM address. we discovered the Java Card has internally a rule about the locality of EEPROM writing address. Consequently, a locality of Java Card objects and data is considerably high.

### 3.3 A High Locality of Heap Area in EEPROM

As mentioned earlier, a traditional Java Card System has only one page-buffer in RAM to write data into EEPROM. The page-buffer has a function for the buffering of just consecutive bytes. In this paper, we suggest the object-buffer that perform a buffering and caching to improve the execution speed of Java Card. The most important and considerable point in order to add caching function to Java Card is a high hitting rate of the caching buffer.

When the wallet class is created by install() method, the wallet object (2011C3A600000000) that have 3 fields is first written in EEPROM, and then, Own-

```

public class wallet extends Applet{
    int balance;
    int withdraw;
    OwnerPIN pin;
    wallet (){ // constructor
        pin = new OwnerPIN(3, 8); // create OwnerPIN(trylimit, Pinsize) object
    }
    initialize(){
        balance = 90;
    }
    withdraw(){ // method
        withdraw = 50;
        balance = balance - withdraw;
    }
}
    
```

} global variables  
→ reference class

**Fig. 5.** wallet applet that has 3 methods and 3 fields; when the wallet applet is created by install() method, OwnerPIN object also is created in wallet() constructor

erPIN object (20111E69000308) that assigned 0045 as an objectID is created and written in EEPROM. After the OwnerPIN object created, Java Card writes the objectID (0045) as pin reference field of the wallet object (2011C3A600000045). After the wallet applet is created, a method such as initialize() and withdraw() generally would be invoked. In figure 4, initialize() method is to change the value of balance field into 100. After this operation, the content of the wallet object is 2011C3A690000045. withdraw() method also changes the field value of withdraw and balance into 50 and 40 separately. At this time, the content of the wallet object is 2011C3A640500045.

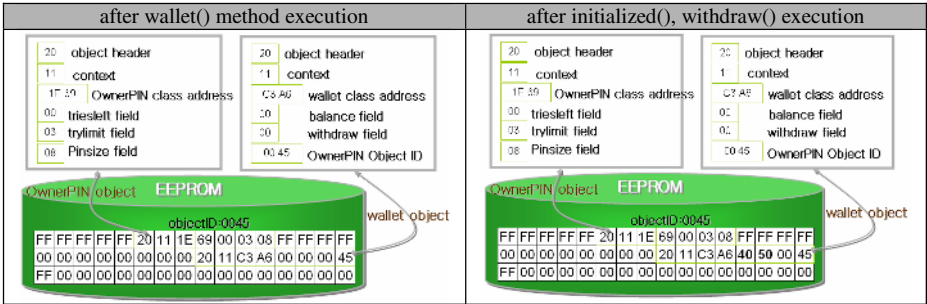


Fig. 6. The creation process of the wallet applet and the OwnerPIN object in EEPROM and the process of the changing localized-fields and rewriting them

Figure 5 and 6 showed several EEPROM writing processes from the creation of wallet applet to the execution of methods such as initialize() and withdraw(). If Java Card just performs these processes by using one page-buffer above-mentioned, it might spends much time in writing and changing localized-data like above example.

## 4 Our Changed Java Card with a Fast Execution Speed

### 4.1 New Transaction Algorithm with T\_Buffer in RAM

As mentioned in the related works, smart cards including a Java Card supports a transaction mechanism by saving old\_values in EEPROM. the number of EEPROM writing in order to support the transaction is about 75 to 80 percent of the total number of EEPROM writing. EEPROM writing is typically more than 1,000 times slower than writing to RAM. It makes also Java Card much more slow and inefficient.

We suggested new transaction mechanism using RAM, not EEPROM in this paper. If such tearing such as power loss happens in the middle of a transaction, all data after transaction began should be ignored. If T\_Buffer area to save old\_values places in RAM, in case of power loss, RAM is automatically reset. It means the preservation of old\_values.

Figure 7 shows the transaction mechanism of a traditional Java Card. After a transaction begin, if tearing such as power loss occurs, Java Card restore data involved in the transaction to their pretransaction(original) values the next time the card is powered on. To do this, Java Card must store all old\_values in T\_Buffer in EEPROM whenever Java Card writes some data in EEPROM.



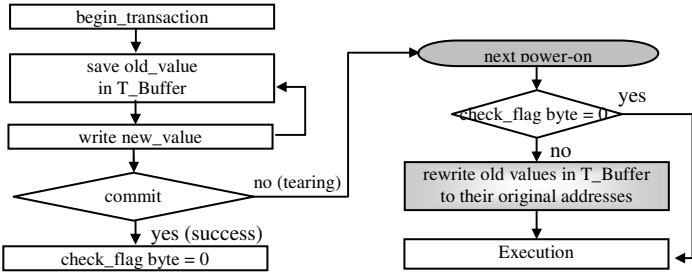


Fig. 7. The transaction mechanism with T\_buffer in EEPROM of a traditional Java Card

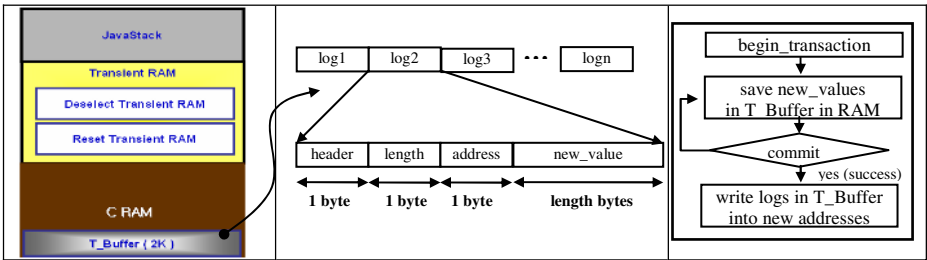


Fig. 8. RAM structure to support our changed transaction mechanism, the structure of our T\_buffer and our transaction mechanism with T\_buffer in RAM of a traditional Java Card

In this paper, we suggest that T\_Buffer to support a transaction is in RAM in order to reduce EEPROM writings. Our T\_buffer in RAM saves all new\_values that will be written in EEPROM after a transaction began. Our T\_Buffer also could have many logs until a transaction commit. Figure 8 below shows the structure of T\_Buffer. Each log entry consists of four fields. The length field is the number of bytes of old data. The address field is original data in EEPROM. The last old\_data field is old data bytes.

**4.2 Our Object-Buffer Based on Java Card Objects with a High Locality**

In chapter 3, we explained how to write data in EEPROM by using one page buffer in a traditional Java Card. It is the one of causes of a Java Card’s performance drop in

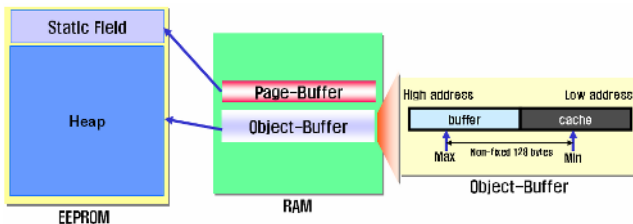
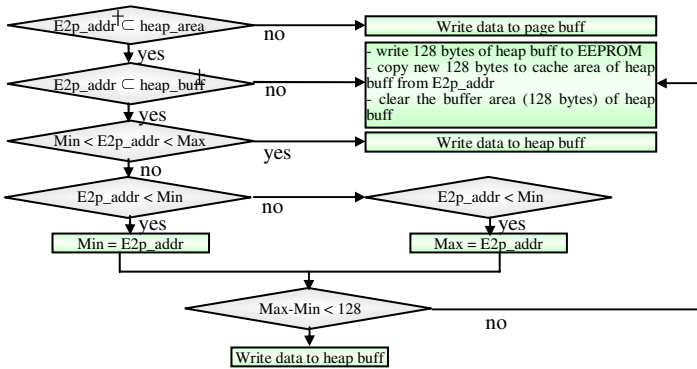


Fig. 9. The heap-buffer that is consisted in 2 part; the buffer and cache. The data between Min and Max can be written to EEPROM at a time.

company with the transaction mechanism of a traditional Java Card. We discovered that all objects and data that the Java Card creates during the execution has a high locality. It means that an additional caching function makes the number of EEPROM writing go down. For these reasons, we developed new Java Card with two page buffer in RAM; one is the existing page buffer for non-heap area, another (object-buffer) is for heap area in EEPROM. The heap area is where objects created by Java Card are allocated.

Figure 10 below shows the main algorithm using the object-buffer and page-buffer. The writing of non heap-area is performed with the existing page buffer. The writing of heap-area is executed with the object-buffer. When the Java Card writes data related to Java Card objects into heap area of EEPROM, the first operation is to get 128 bytes lower than the address that will be written and to copy them to the cache area of the object-buffer. Next, the buffer area(128-byte) of the object-buffer is cleared. Two points, Max and Min have the highest and lowest points that are written after Java Card get new 256 bytes to the object-buffer. the gap between them continually is checked in order to write the heap buffer to EEPROM. Max and Min are non-fixed points to raise the efficiency of the heap buffer. The reason why the gap between Max and Min is 128 bytes is that our target chip, CalmCore16, supports the EEPROM writing of 128 bytes at a once.



**Fig. 10.** The object-buffer algorithm that checks continually the Min and Max points to write the object-buffer to EEPROM when Java Card writes data to heap area. (†E2p\_addr : the EEPROM address that data will be written, ‡ heap\_buff(object-buff) : our new heap buffer with caching and buffering function for just heap area in EEPROM).

## 5 Evaluation of Our Approach

The key of our approach is improve an execution speed of the Java Card by reducing the number of EEPROM writing. The main idea is also that EEPROM writes are typically more than 1,000 times slower than writes to RAM. One of the analyzed results of a traditional Java Card is that Java Card has a inefficient transaction mechanism to guarantee an atomicity and page-buffer algorithm to write data to EEPROM regardless of the high locality of Java objects. For this reason, we developed new transaction mechanism and new page buffer algorithm.

In our approach, to get more precise figure in the real Java Card, we made an experiment with CalmCore16 MCU [14], SAMSUNG MicroController for smart card.

Figure 11 below shows the comparison between a traditional Java Card and our changed Java Card in regard to the number of EEPROM writing and the execution speed. First of all, the number of EEPROM writing is reduced by about 80% by using the T\_Buffer and the object buffer in RAM.

Applets	Traditional	Our Approach	Reduced
ChannelDemo	7552	1586	79%
JavaLoyalty	7291	1322	82%
JavaPulse	22712	4537	80%
ObjDelDemo	16416	3025	82%
PackageA	9685	2000	79%
PackageB	7698	1406	82%
PackageC	3439	745	79%
PhotoCard	6737	1400	79%
RMIDemo	6119	1261	79%
Wallet	5641	1190	79%
EMV small Applet	6721	1419	79%
EMV Large Applet	11461	2433	79%
<b>Average</b>			<b>80%</b>

Applets	Traditional	Our Approach	Reduced
ChannelDemo	76140	49438	35%
JavaLoyalty	72703	46187	36%
JavaPulse	232100	150359	35%
ObjDelDemo	159420	99157	38%
PackageA	90530	56375	38%
PackageB	74859	49937	33%
PackageC	32743	20907	36%
PhotoCard	64608	41407	37%
RMIDemo	57328	36235	34%
Wallet	57140	37438	37%
EMV small Applet	61766	38859	37%
EMV Large Applet	119812	79422	34%
<b>Average</b>			<b>36%</b>

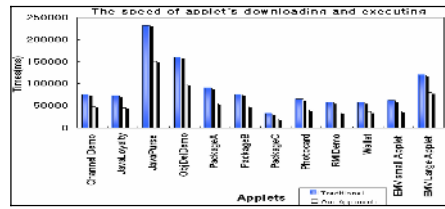
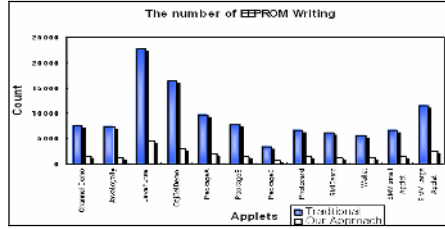


Fig. 11. The comparison between a traditional Java Card and our changed Java Card with regard to the number of EEPROM writing and the execution speed

Components	Traditional	Our Approach	Difference
Initialize	1485	1688	-203
Select Install	6281	3812	2469
CAP Begin	1234	485	749
Header	3562	2156	1406
Directory	3969	2344	1625
Import	2875	1640	1235
Applet	3250	1922	1328
Class	2203	1484	719
Method	11266	8641	2625
StaticField	2297	1469	828
ConstantPool	6781	4984	1797
ReferenceLocation	9141	4719	4422
CAP End	625	422	203
Create Applet	2171	1672	499
Total	57140	37438	19702

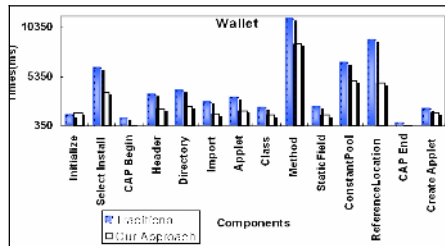


Fig. 12. The comparison between a traditional Java Card and our changed Java Card in regard to Wallet applet's downloading and execution speed per each component

One applet consists of over 11 components that include all information of one applet package. We also produced downloading results about each component. Basically, when Java Card installer downloads one applet, the component that takes a long time is the referencelocation component. The reason is that both are related to the resolution of indirect references during the downloading. our approach almost reduced the downloading time of the referencelocation by 50%.

## 6 Conclusion and Future Work

Java Card technology is already a standard for smart cards and SIM cards [11, 15]. A Java language is basically slower than other languages. The card platforms also have a heavy hardware limitation. In spite of a Java's slow speed, the reasons why Java Card technology is selected as a standard are a post-issuance and a platform independence. When Java Card downloads new application, a post-issuance generally spends a lot of time [10, 11].

In this paper, we have proposed the method to reduce the number of EEPROM writing with new robust transaction mechanism and new object-buffer based on the high locality of Java Card objects. It also makes Java Card more fast. With our approach, the number of EEPROM writing and the downloading speed reduced by 80% and 35% separately. It also enables an application to be downloaded more quickly in the case of an application sent to a mobile phone via the GSM network (SIM). This technology will be applied to embedded systems such as KVM, PJAVA, CLDC with a Java Technology.

## References

1. Sun Microsystems, Inc. JavaCard 2.2.1 Virtual Machine Specification. Sun Microsystems, Inc. URL: <http://java.sun.com/products/javacard> (2003).
2. Sun Microsystems, Inc. JavaCard 2.2.1 Runtime Environment Specification. Sun Microsystems, Inc. URL: <http://java.sun.com/products/javacard> (2003).
3. Chen, Z. Java Card Technology for Smart Cards: Architecture and programmer's guide. Addison Wesley, Reading, Massachusetts (2001).
4. W.Rankl., W.Effing., : Smart Card Handbook Third Edition, John Wiley & Sons (2001).
5. James Gosling, Bill Joy, Guy Steele, and Gilad Bracha. : The Java Language Specification, Second Edition. Addison-Wesley, <http://java.sun.com/docs/books/jls/index.html> (2001).
6. Marcus Oestreicher, Ksheerabdh Krishna. : USENIX Workshop on Smartcard Technology, Chicago, Illinois, USA, May 10–11, 1999.
7. M. Oestreicher and K. Ksheerabdh, "Object Lifetimes in JavaCard,," Proc. Usenix Workshop Smart Card Technology, Usenix Assoc., Berkeley, Calif., (1999) 129–137.
8. Michael Baentsch, Peter Buhler, Thomas Eirich, Frank H6ring, and Marcus Oestreicher, IBM Zurich Research Laboratory, Java Card From Hype to Reality (1999).
9. Pieter H. Hartel , Luc Moreau. : Formalizing the safety of Java, the Java virtual machine, and Java card, ACM Computing Surveys (CSUR), Vol..33 No.4, (2001) 517-558.
10. M.Oestreicher, "Transactions in JavaCard,," Proc. Annual Computer Security Applications Conf., IEEE Computer Society Press, Los Alamitos, Calif., to appear, Dec. 1999.
11. Kim, J. S., and Hsu, Y.2000. Memory system behavior of Java programs: methodology and analysis. In Proceedings of the ACM Java Grande 2000 Conference, June.
12. 10. <http://www.gemplus.com>. : OTA White Paper. Gemplus (2002).
13. the 3rd Generation Partnership Project. : Technical Specification Group Terminals Security Mechanisms for the (U)SIM application toolkit. 3GPP (2002).
14. MCULAND, <http://mculand.com/e/sub1/s1main.htm>.
15. X. Leroy. Bytecode verification for Java smart card. Software Practice & Experience, 2002 319-340
16. SAMSUNG, <http://www.samsung.com/Products/Semiconductor>
17. SIMAlliance, <http://www.simalliance.org>.

# Intelligent Object Extraction Algorithm Based on Foreground/Background Classification

Jhing-Fa Wang, Han-Jen Hsu, and Jyun-Sian Li

Department of Electrical Engineering, National Cheng Kung University,  
No.1, Ta-Hsueh Road, Tainan, Taiwan  
wangjff@csie.ncku.edu.tw, hjhsu@icwang.ee.ncku.edu.tw  
<http://icwang.ee.ncku.edu.tw>

**Abstract.** In this paper, we propose an intelligent object extraction algorithm based on foreground/background classification. The proposed algorithm can offer the users more friendly interface for object extraction from image without unnecessary steps. After the interactive steps from user (marking the foreground and background parts), the wanted object is extracted from the background automatically. The proposed algorithm processes the input image by watershed to produce the regions. Then, the regions are labeled after marking parts of regions. We also introduce an implementation of hierarchical queues to store the unlabeled regions. The classification of foreground and background will generate the final image with selected object. In our experimental results, the proposed algorithm provides the output image with high efficiency. The wanted object is generated after user marking the foreground and background parts less than one second. In addition, the application of this work also can be used in image synthesis or object removal in other fields of image processing.

**Keywords:** Object extraction, image editing tool, image segmentation.

## 1 Introduction

In traditional researches, image segmentation means to detect the boundaries in digital image. However, the boundaries of whole image may contain the boundaries of textures or the inner structure of object in the foreground. In this work, we need to find the contour of wanted object which separates the image into only two parts-foreground and background. Therefore, object extraction can be considered as a binary labeling problem.

The related works include boundary-based methods and region-based methods. For boundary-based methods, they tried to approximate the pre-assigned curves near the object to the object contour, such as intelligent scissor [1], image snapping [2], and Jetstream [3]. The users need to draw the curves which enclose the whole object. This disadvantage of these methods is high complexity in user interface. The user needs to draw the shape of object explicitly.

In order to reduce the redundant steps in user interface, several researches are mentioned to improve the previous works in boundary-based methods. Recently, the re-

gion-based methods are proposed. The accuracy is increased without as much efforts as that in boundary-based methods. The primitive concerns of this problem are two points. One problem is using less effort to acquire more accurate result. Another problem is the detail information of the object also needed to be preserved. Sun *et al.* proposed a smart method “Poisson Matting” [4] focused on preserving fine features of object, such as hairs and feathers. The object can be cut from original image and pasted on another target image. Rother *et al.* proposed “GrabCut” [5] to achieve foreground extraction using iterated Graph Cuts. The target object is extracted by dragging a square window around the target. Another work “Lazy snapping” [6], is presented by drawing the foreground and background parts at first. The user interface is similar to our system. An example of our proposed algorithm is provided in Fig. 1. The user draws two kinds of lines, green lines for foreground seeds and yellow lines for background seeds, in the input image in Fig. 1(a). The wanted object is acquired easily in the output image shown in Fig. 1(b). Based on foreground and background classification, the regions belonged to wanted object are given as foreground label ( $F$ ). The other remaining regions are classified to background label ( $B$ ). However, the object also can be labeled as background by opposite assignments of foreground and background. The obtained result is the input image for object removal.

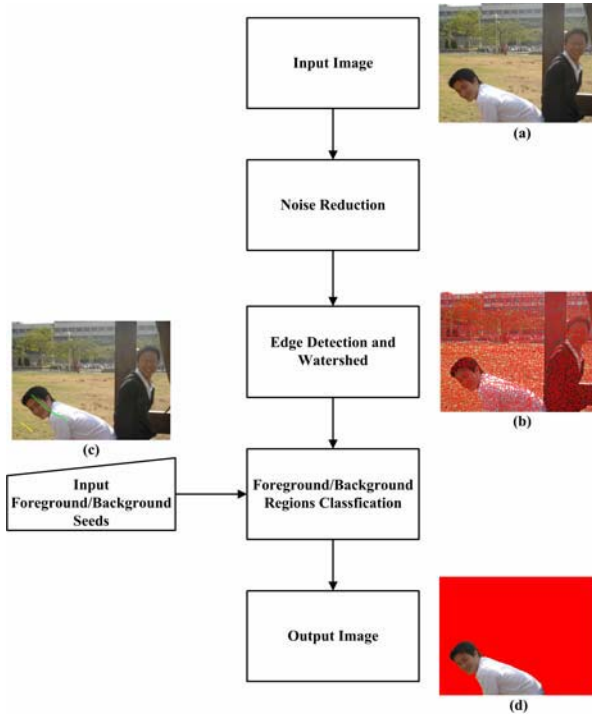
The organization of this paper is shown as follows. In Section 2, we will describe our algorithms in detail. In Section 3, the experimental results are shown to provide subjective measurement. Finally, the conclusion and future work are provided in Section 4.



**Fig. 1.** The example of our proposed algorithm

## 2 Proposed Intelligent Object Extraction Algorithm

The flow diagram of our proposed algorithm is shown in Fig. 2. At first, the input image in Fig. 2(a) is pre-processed to reduce additive noise. The edge detection and watershed algorithm are applied to produce many small regions shown in Fig. 2(b). Then, the user marks the image by foreground (green lines) and background (yellow lines) markers, respectively, as shown in Fig. 2(c). Once the user marks the image, according to the two sets of markers, two sets of regions are labeled as  $F$  and  $B$ , respectively. And the non-marked regions are defined as unlabeled.



**Fig. 2.** The flow diagram of our proposed algorithm

After all, the unlabeled regions are classified into foreground or background to generate the final image as shown in Fig. 2(d). The detail description of the proposed algorithm is shown as follows.

## 2.1 Noise Reduction

The pre-processing is applied to remove the noise which may affect the output result. We use median filter and mean filter to reduce the noise [7]. The median filter is applied first to avoid the operation of averaging at impulse noise in mean filter.

## 2.2 Edge Detection and Watershed

After noise reduction, edge detection and watershed are used to segment the input image into large number of regions. In order to enhance the luminance and color variation near the boundary of object, we adopt a simple and efficient method which incorporates the morphological gradient of luminance and color in  $L^*a^*b^*$  color space from [8]. The gradient value is decided by the erosion and dilation. The color space transformation of RGB to  $L^*a^*b^*$  is described from [9]. The RGB values to XYZ(D65) is shown as in (1).

$$\begin{bmatrix} X_{D65} \\ Y_{D65} \\ Z_{D65} \end{bmatrix} = \begin{bmatrix} 0.3935 & 0.3653 & 0.1916 \\ 0.2124 & 0.7011 & 0.0865 \\ 0.0187 & 0.1119 & 0.9582 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1)$$

The pixel values in L\*a\*b\* color space transformation is provided as in (2).

$$L^* = \begin{cases} 116 \left( \frac{Y}{Y_n} \right)^{1/3} - 16, & \left( \frac{Y}{Y_n} \right) > 0.008856 \\ 903.3 \left( \frac{Y}{Y_n} \right), & \left( \frac{Y}{Y_n} \right) \leq 0.008856 \end{cases}$$

$$a^* = 500 \times \left[ f \left( \frac{X}{X_n} \right) - f \left( \frac{Y}{Y_n} \right) \right] \quad (2)$$

$$b^* = 200 \times \left[ f \left( \frac{Y}{Y_n} \right) - f \left( \frac{Z}{Z_n} \right) \right]$$

$$\text{where } f(t) = \begin{cases} t^{1/3}, & t > 0.008856 \\ 7.787t + \frac{16}{116}, & t \leq 0.008856 \end{cases}$$

We have used D65 as the CIE L\*a\*b\* reference white point. Thus, the constant values  $X_n$ ,  $Y_n$ , and  $Z_n$  are equal to 0.9504, 1.0, and 1.0889, respectively.

After edge detection, we use the watershed algorithm from [7] to chunk the image into many regions. We then construct the region adjacency graph (RAG) [10]-[11], which represents the relation between each region and its neighborhood. With this useful step, we can extract the object more efficiently. The RAG is defined as an undirected graph,  $G = (V, E)$ , where  $V = \{1, 2, 3, \dots, k\}$  is the set of graph nodes,  $k$  is the number of regions obtained from watershed, and  $E \subset V \times V$  is the set of graph edges. Each region is represented by a graph node and there exists a graph edge  $(x, y)$  if the two graph nodes  $x$  and  $y$  are adjacent.

A weight of each graph edge stands for the regional color distance of the two adjacent regions, as shown in (3).

$$\text{RCD}(x, y) = \|C(x) - C(y)\| \quad (3)$$

where  $C(\bullet)$  denotes the mean color vector of a region in L\*a\*b color space,  $x \in V$ ,  $y \in V$ , and  $(x, y) \in E$ .

### 2.3 Input Foreground/Background Seeds

The all regions are all unlabeled regions before this step. In this step, we mark the image with foreground and background seeds. More drawing the lines of the foreground and background will lead more exact result. However, our algorithm needs less effort to generate the wanted object. The foreground seeds are selected in green and the background seeds are selected in yellow, respectively. Once the user marks the image, foreground and background marked regions are labeled as  $F$  and  $B$ , respectively. Therefore, the marked regions are the labeled regions inevitably. And the non-



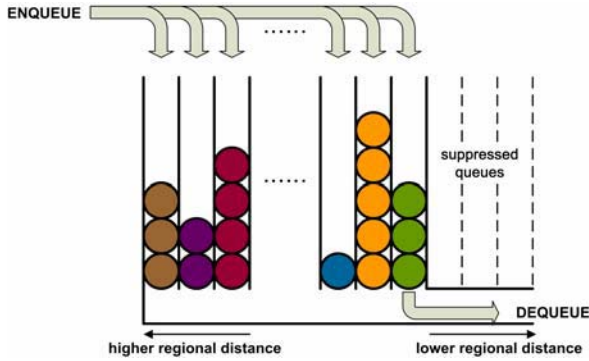


Fig. 3. The notation diagram of hierarchical queues

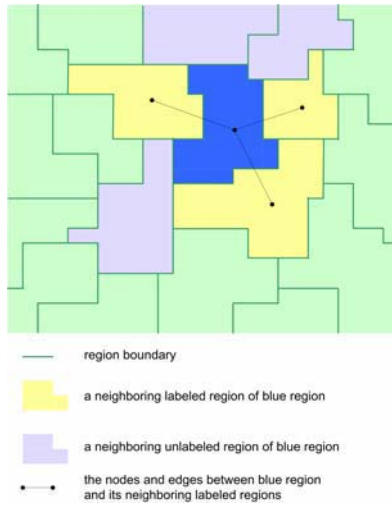


Fig. 4. The generic notation diagram of region labeling

marked regions are regarded as unlabeled regions. In the next step, the remaining non-marked regions are processed by *F/B* classification.

**2.4 F/B Classification (Region Labeling)**

In this Section, the other unlabeled regions which are not belonged to user-defined foreground and background regions are labeled in this Section. *F/B* classification (Region labeling) is the most important step in our algorithm, which affects the quality of final result. The pseudocode of region labeling is shown in Table. 1.

In our implementation, we adopt the hierarchical queues shown in Fig. 3 from [12]. In each queue of hierarchical queues, the regions with the same index number are put in the same queue. For example, there are three green balls in the same queue which means three regions with the same index number. This data structure is used in image

segmentation originally. However, this mechanism is also suitable for region labeling. It is composed of two steps: Initialization of the Hierarchical Queues and the Flooding Step. The generic notation diagram is shown in Fig. 4, the blue region is denoted as  $\mathbf{A}$ , in initialization of hierarchical queues. The regional color distance between region  $\mathbf{A}$  and neighboring labeled regions is used to decide which index number of region  $\mathbf{A}$  in hierarchical queues is. On the other hand, the blue region is denoted as  $\mathbf{R}$  in the flooding step. Therefore, the regional color distance between region  $\mathbf{R}$  and neighboring labeled regions is used to decide which label of region  $\mathbf{R}$  is. The detail descriptions of these two steps are shown as below.

**Table 1.** The pseudocode of region labeling in our proposed algorithm

Step	Description
<b>Initialization of the Hierarchical Queues:</b>	
For each labeled region $\mathbf{L}$	
For each neighboring region $\mathbf{A}$ of $\mathbf{L}$	
if $\mathbf{A}$ is a unlabeled region outside of the hierarchical queues enqueue $\mathbf{A}$ into the hierarchical queues according to its index number.	
<b>Flooding Step:</b>	
Repeat the following steps until the all hierarchical queues are empty:	
1.	dequeue a region $\mathbf{R}$ from the hierarchical queues from the lowest index number;
2.	region $\mathbf{R}$ has at least one labeled region in its neighborhood. It is assigned to the same label with its neighboring labeled region which has the smallest distance from $\mathbf{R}$ ;
3.	the neighboring regions of $\mathbf{R}$ that have not been labeled and are still outside the hierarchical queues are enqueued into the hierarchical queues with the index number not lower than the index number of $\mathbf{R}$

### Initialization of the Hierarchical Queues

The initialization of region labeling is enqueued the neighboring unlabeled regions of the user marked regions into the hierarchical queues. Each neighboring unlabeled region ( $\mathbf{A}$ ) of user marked regions is enqueued into the hierarchical queues according to its index number as in (4). The index number of a region is denoted as  $q$ .

$$q = \text{floor}(\min_m(\text{RCD}(\mathbf{R}_i, \mathbf{A})) + 0.5) \quad \text{where } i = 1 \text{ to } m \quad (4)$$

where  $(\mathbf{R}_i, \mathbf{A}) \in E$  and  $m$  is the number of the labeled regions adjacent to  $\mathbf{A}$ .

### Flooding step

After the initialization of the hierarchical, we start to dequeue the regions from the queue with the lowest index number. Any queue which is empty will be suppressed

and no longer be enqueued. The hierarchical queues are processed until all the queues are empty. The flooding step is similar to the initialization of the hierarchical queues. Each region in the hierarchical queues is dequeued and compared the similarity with the neighboring labeled regions. The labeled regions may contain the user marked regions and the regions which are already labeled in this step. As shown in Table. 1, after we dequeue a region  $\mathbf{R}$  from the hierarchical queues, we have to find a neighboring labeled region of  $\mathbf{R}$  which is most similar to  $\mathbf{R}$  as in (5).

$$\mathbf{R}^* = \arg \min_{\mathbf{LR}} \text{RCD}(\mathbf{LR}, \mathbf{R}), \quad \text{where } \begin{cases} \mathbf{LR} \in N(\mathbf{R}) \\ \mathbf{R}^* \in N(\mathbf{R}) \end{cases} \quad (5)$$

$\mathbf{R}^*$  is the most similar region to  $\mathbf{R}$ , and  $N(\mathbf{R})$  denotes the neighboring labeled region of  $\mathbf{R}$ . Then,  $\mathbf{R}$  is assigned to the same label as  $\mathbf{R}^*$ .

On the other hand, the neighboring unlabeled regions ( $\mathbf{B}$ ) of region  $\mathbf{R}$  which is still out of the hierarchical queues is enqueued into the  $q$ -th queue as in (6).

$$t = \underset{n}{\text{floor}}(\min(\text{RCD}(\mathbf{R}_j, \mathbf{B})) + 0.5) \quad \text{where } j = 1 \text{ to } n \quad (6)$$

$$\begin{cases} q = t, & \text{if } t \geq z \\ q = z, & \text{otherwise} \end{cases} \quad \text{where } z \text{ is the index of } \mathbf{R}$$

where  $\mathbf{R}_j$  is labeled,  $(\mathbf{R}_j, \mathbf{B}) \in E$ , and  $n$  is the number of the labeled regions adjacent to  $\mathbf{B}$ .

After this, if the queue with the same index number of  $\mathbf{R}$  is empty, it will be suppressed. In the later processing, if we obtain a region which will be enqueued into the suppressed queue, we put the regions into the lowest un-suppressed queue. Finally, until the whole regions are labeled (the hierarchical queues is empty), the wanted objects are extracted from the input image.

### 3 Experimental Results

We provide some experimental results in this Section. The experimental results show our proposed algorithm can produce excellent output images. The test image from Kodak test images ‘‘parrot’’ is chosen by the user shown in Fig. 5(a). After our proposed algorithm, the red parrot is extracted from input image shown in Fig. 5(b). The use only needs to draw little lines of foreground and background. Another example is provided in Fig. 6. The wanted lighthouse is extracted from the image shown in Fig. 6(b). The test image from [13] in Fig. 7 shows three chairs on the grass. The red chair is chosen to be the target object. We draw two points in green and one line in yellow as background. The middle red chair is extracted from the image shown in Fig. 7(b).

Besides of natural images, we also use indoor image to test our proposed algorithm. In Fig. 8(a), we use the test image from [14]. The operation of marking the foreground and background are reversed to generate the opposite result. The wanted object is selected as background which we want to remove from the image. In another work, ‘‘object removal’’, the result shown in Fig. 8(b) is used to be the input image.

We take another photograph as shown in Fig. 9. The left man is chosen to be the object which we want to remove. The output result is shown in Fig. 9(b). The computation time analysis is given in Table. 2. The computation time costs most time in edge detection and watershed algorithm. The final result is obtained immediately after the user drawing. The simulation environment is on AMD 1.7G with 1GB of RAM and implemented in C++.



Fig. 5. The experimental result for object extraction



Fig. 6. The experimental result for object extraction



Fig. 7. The experimental result for object extraction

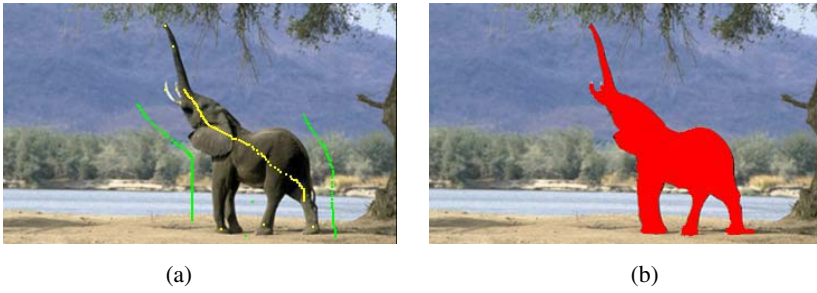


Fig. 8. The experimental result for object removal



Fig. 9. The experimental result for object removal

Table. 2. The computation time analysis of our proposed algorithm

Image no.	Image Resolution	Noise Reduction	Edge Detection	Watershed	F/B Classification
1	768*512	1.390 sec	5.469 sec	2.250 sec	0.031 sec
6	768*512	1.421 sec	5.406 sec	2.172 sec	0.016 sec
7	768*512	1.406 sec	5.422 sec	2.234 sec	0.032 sec
8	352*234	0.282 sec	1.141 sec	0.453 sec	0.001 sec
9	450*339	0.500 sec	2.078 sec	0.813 sec	0.001 sec
10	352*211	0.266 sec	1.031 sec	0.422 sec	0.001 sec
11	352*264	0.329 sec	1.266 sec	0.484 sec	0.001 sec

### 4 Conclusion and Future Work

In this work, we propose an intelligent object extraction algorithm based on foreground/background classification. The regions after watershed are classified into

foreground and background. The hierarchical queues are implemented to increase the efficiency. The user interface is easy to use, even the general users without the tips in image processing can acquire the wanted object. The proposed algorithm can produce good results in real-time.

Our future work is preserving the fine features on the boundary of the object. Currently, the fine features of the object need to be selected as foreground to ensure the excellent result. We are going to improve the weakness of our proposed algorithm. To develop the smart camera system, we also plan to integrate with object removal from our previous work.

## References

1. Mortensen, E. N., Barrett, W. A.: Toboggan-based intelligent scissors with a four parameter edge model. In *Proceedings of CVPR'99*. (1999)
2. Gleicher, M.: Image snapping. In *Proceedings of ACM SIGGRAPH'95*. (1995)
3. Perez, P., Blake, A., Gangent, M.: Jetstream: Probabilistic contour extraction with particles. In *Proceedings of ICCV 2001*. (2001)
4. Jian Sun, Jiaya Jia, Chi-Keung Tang, Heung-Yeung Shum: Poisson matting. *ACM Transactions on Graphics (TOG)*, Volume 23 Issue 3 (2004)
5. Carsten Rother, Vladimir Kolmogorov, Andrew Blake: "GrabCut": interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (TOG)*, Volume 23 Issue 3 (2004)
6. Yin Li, Jian Sun, Chi-Keung Tang, Heung-Yeung Shum: Lazy snapping. *August ACM Transactions on Graphics (TOG)*, Volume 23 Issue 3 (2004)
7. Rafael C. Gonzalez, Richard E. Woods.: *Digital Image Processing*. Prentice-Hall, Inc. (2002)
8. Hai Gao, Wan-Chi Siu, Chao-Huan Hou: Improved Techniques for Automatic Image Segmentation. *Circuits and Systems for Video Technology*, *IEEE Trans. on* Volume 11, Issue 12, pp. 1273 - 1280 (2001)
9. J. M. Kasson, W. Plouffe: An Analysis of Selected Computer Interchange Color Spaces. *ACM Transactions on Graphics*, Vol. 11, No. 4, October, Pages 373-405 (1992)
10. D. Ballard and C. Brown: *Computer Vision*. Englewood Cliffs, NJ: Prentice-Hall (1982)
11. X. Wu: Adaptive split-and merge segmentation based on piecewise least-square approximation. *IEEE Trans. Pattern Analysis Machine Intelligence* Vol. 15, Page 808-815 (1993)
12. Meyer, F.: Color Image Segmentation. *Image Processing and its Applications*, International Conference 303 - 306 (1992)
13. I. Drori, D. Cohen-Or, H. Yeshurun: Fragment-based image completion, in *ACM Trans. Graphics (TOG)*, vol.22, no. 3, pp. 303-312 (2003)
14. J. Jia, and C. K. Tang: Inference of segmented color and texture description by tensor voting. *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, June (2004)

# Thin Client Based User Terminal Architecture for Ubiquitous Computing Environment

Tatsuo Takahashi<sup>1</sup>, Satoshi Tanaka<sup>1</sup>, Kenichi Yamazaki<sup>1</sup>, and Tadanori Mizuno<sup>2</sup>

<sup>1</sup> Network Laboratories, NTT DoCoMo, Inc., 3-5 Hikari-no-oka, Yokosuka 239-8536, Japan  
{tatsuo, satoshi, yamazaki}@netlab.nttdocomo.co.jp

<sup>2</sup> Faculty of Information, Shizuoka University, 3-5-1 Jo-hoku, Hamamatsu 432-8011, Japan  
mizuno@cs.inf.shizuoka.ac.jp

**Abstract.** In this paper, the authors examine the use of thin client based user terminals to realize the RFID tag based ubiquitous computing environment. The ubiquitous service targeted is not information retrieval via RFID but the user observation service based on environment perception. Thus the user terminal must ensure service consistency even when the communication link to the server is disconnected. In order to achieve this, the authors propose an event cache mechanism that stores predicted event conditions and the corresponding reactions. A prototype and evaluation results are also described.

## 1 Introduction

In the ubiquitous computing environment advocated by Marc Weiser [1], various objects surrounding us will have computing ability. They will recognize the situation of the user and his/her surroundings, select the best service, and offer it without error or intrusion. In such an environment, the user will not have to know how to use a computer nor recognize the existence of the system. Therefore, the conventional wisdom is that user-carried terminals such as note PCs and PDAs will lose their significance in the ideal ubiquitous computing environment. However, the authors consider that the user carried terminals (hereafter called user terminal) will be needed for the time being, in order to evaluate and collect the ubiquitous services provided, offer an exclusive use actuator (which prevents interference by a third party), and follow the moving user to collect his/her situation continuously.

From such a standpoint, the authors examine the architecture of the user terminal and computing device controlling it with regard to implementing the ubiquitous computing environment.

The authors propose a thin client based user terminal that is realized as a cellular phone with an RFID tag reader. No specific RFID format is assumed.

## 2 The User Terminal for the Ubiquitous Computing Environment

### 2.1 Assumed Ubiquitous Computing Environment

Figure 1 illustrates the ubiquitous computing environment assumed in this paper. In this environment, computing devices in cyber-space capture the real-space from RFID

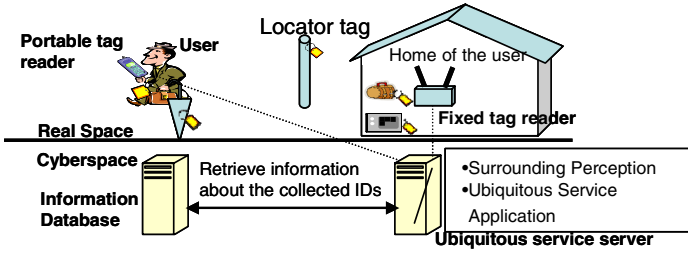


Fig. 1. Assumed ubiquitous computing environment

tags as detected by tag readers. RFID tags are attached to various objects. Locator tags are special RFID tags that identify places not objects. Locator tags will be attached to train stations, major buildings, and intersections. Each RFID tag stores just a unique identifier. Tag readers collect the IDs of the RFID tags in their range, and forward this information to the computing devices (hereafter called server) via the Internet and/or mobile communication networks. The server passes the ID to an information database, and retrieves information (name, role, color, owner, and other attributes) of the object. The server infers the user’s surroundings from the collected information and offers various ubiquitous services as appropriate.

Though the simple information search service based on RFID tags is also called ubiquitous service, the authors address the observation and life assistance services based on RFID-based environment perception. Also, in this paper, the authors focus on services for the mobile user. Examples are Lost Property Notification service[2] which gives real-time notification when user drops a carried object or it seems to have been stolen, and Shopping List service [2] which is a reminder service for when the user seems to have forgotten to purchase something (or goes to the shop).

## 2.2 User Terminal Requirements for the Assumed Environment

In the ubiquitous computing environment assumed, each user should carry his/her user terminal in order to satisfy the following requirements.

### (1) Platform for the portable tag reader

Because the detection range of most tag readers is limited to about ten meters even if the UHF band is used, it is difficult to cover all areas completely by using fixed tag readers. If the server cannot detect the event that should trigger the service, user satisfaction will be degraded. Thus, in the assumed environment, each mobile user always carries a portable tag reader to provide real-time the event detection around him/her; the collected RFIDs are transferred to the server over the mobile communication network.

In order to control the portable tag reader, collect the tag information, and transfer the information via the mobile communication network, some kind of platform is required. In the assumed environment, the user terminal acts as this platform.

### (2) Subordinate control for the server

It is difficult to provide the service that the user desires all of the time even if the server perceives the surrounding from a huge amount of information and an advanced



inference engine is used. Therefore, it is necessary to provide a service evaluation function, and feed the user's feelings back to the server. Moreover, it is considered that the user must make the final decision on important matters such as those related to the user's life and property. The user terminal must offer support functions to achieve these goals.

### (3) Actuation function

In order for the ubiquitous service to reach the user, the mechanism of service actuation is required. If shared terminals and robots, etc. are used as the actuator, information related to the user's privacy and security will be leaked in public spaces. Therefore, the user's terminal should become an exclusive actuator that provides adequate privacy.

## 2.3 General Requirements for the User Terminal

As described in the previous section, the user terminal is needed in the environment considered here. In this section the general requirements for this user terminal are described.

- **Portability:** Because each user always has to carry the user terminal, it should be light and small.
- **Low power consumption:** Because the user terminal must always be active when the ubiquitous service is required, it should have low power consumption and should run for long periods without battery change.
- **Low cost:** This seems to be a fundamental requirement for every user.

## 2.4 Other Work Related to the User Terminal

In this section the authors describe the previous research on user terminals for the ubiquitous computing environment.

Project Oxygen of MIT uses the user terminal called H21 [3]. Unlike the environment assumed in this paper, Project Oxygen uses image and voice to gauge the user's surrounding. Accordingly, H21 has a camera and a microphone. Additionally, the perception and offering service is provided by H21 itself, so a huge amount of processing performance is required which increases the electric power consumption and cost.

The ubiquitous communicator [4] is a small, light user terminal like a PDA, which has an RFID tag reader. It can read ucode-compliant RFID tags and display the linked information. However, it does not target the RFID-based environment perception services for mobile user described in Section 2.1.

## 3 Thin Client Based User Terminal

As described above, existing research does not satisfy the requirements for the user terminal in the assumed environment. Accordingly, the authors propose the thin client based user terminal; it is a small, light, and low cost terminal (e.g. cellular phone) that

realizes RFID-based surrounding perception and acts as the actuator for the ubiquitous services.

### 3.1 Basic Architecture

The thin client [5][6] is an architecture that concentrates execution, storing, and management of the application on the server, and the client function is limited to HMI(Human Machine Interface) and some part of I/O. Therefore, a simple terminal can realize the functions and the performance of a PC (Personal Computer). In particular, previous research, called mobile thin client [7], introduced a cellular phone-based thin client system that enables PC applications (e.g. Word, Excel, and PowerPoint) to be used through cellular phone.

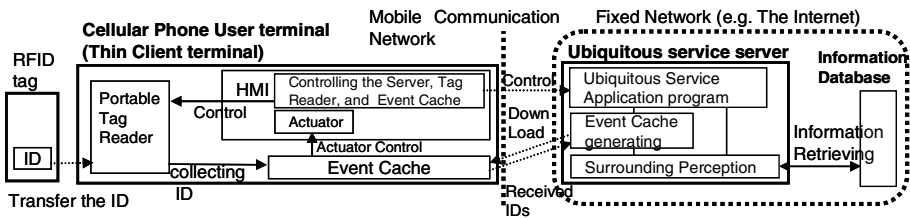


Fig. 2. Architecture of proposed user terminal

In this paper, the authors extended the mobile thin client to realize a cellular phone-based user terminal. Figure 2 illustrates the architecture of the proposed terminal. The HMI function for controlling (start, terminate, correct, selection of critical choices) server ubiquitous service is simply inherited from the mobile thin client. All other functions such as attached portable type tag reader, collecting the RFIDs function, and actuator control functions (BEEP, vibration, backlight blinking, and messaging to the user via the display of the cellular phone) are extensions. The collected RFIDs are transferred to the server via the mobile communication network. The server retrieves the information about the objects from the information database, perceives the user's surroundings from this information, and generates actuator control instructions for the cellular phone. In addition, the proposed architecture offers the function of continuing the ubiquitous services even when the mobile communication network is temporarily disconnected. This mechanism, called event caching, is described in detail in the following section.

### 3.2 Event Caching

The major problem of the thin client system, developed in the authors' previous research, is that when the communication link between the thin client and the server is disconnected, the thin client fails to work. Because the population cover rate of mobile communication network has already reached 100% in Japan, it has far better connectivity compared to the range of fixed tag readers. However, temporary disconnection is common in some environments such as inside subway trains and

areas of heavy network traffic. Because the targeted services described in Section 2.1 are related to the user's life and property, these interruptions should be offset.

The event cache guarantees service continuity. It is achieved by both event prediction in the server and event detection and caching the reaction (i.e. actuator control) in the thin client. After the server perceives the user's current surroundings, it predicts the next event, and then generates the appropriate reaction that should be done when the event occurs. This information is downloaded to the event cache in the user terminal and held until needed or replaced by newer information.

### 3.2.1 Event Prediction

In this section, event predictability is described using the example of the "Lost Property Notification service", a typical ubiquitous computing application described in Section 2.1. It seems easy to judge that the user still has his/her property (object) from the reception by his/her tag reader of the object's RFID tag signal. When the object's signal is no longer received, the ubiquitous application should judge whether the user put the object somewhere on purpose, or he left it somewhere in error, in order to generate the correct reaction. Such processing has to be performed by the server because it requires much processing power and access to a comprehensive information database on the fixed network.

The authors note that the number of objects that should be observed is limited in several ubiquitous applications including the Lost Property Notification service. The server recognizes the current state of each object and decides what kinds of events are possible in the near-term future (e.g. if the user picks up an object, the server predicts that he/she may drop it as a possible next step). Moreover, by using the locator tag described in Section 2.1, the server can recognize the current place of the user; moreover, it is considered that the places that the user will pass are limited and predictable in the short term. It is considered that by using the features of each object (role, price, etc.), the kinds of possible events, and features of the places where an event may occur (public area or private area), the content of the reaction (level and content of warning) that should be executed when the event occurs, can be predicted. The number of events for which reactions are needed is limited because all of the above parameters are limited. After the server identifies the events that are possible in the near-term future and generates the corresponding reactions, it downloads this information into the user terminal. It is a relatively simple task for the thin client terminal to observe the event and execute the appropriate reaction quickly even if the mobile communication network is disconnected.

### 3.2.2 Event Classification

In this section the authors define the states and events. The term *state* means the relationship between the user and the object, and the term *event* means a state transfer. Because of the assumption that the user always carries a tag reader, it is possible to simplify the definition to three states and three events as described below.

WAITS\_FOR state is the state in which the RFID reader cannot receive the signal from the target RFID tag. In the real space, this state corresponds to the situation in which the user is not carrying nor encounters the object. When the reader finds the targeted RFID, it raises the FOUND event and the DETECTED state is entered.

DETECTED state is the state in which the RFID reader first receives a signal from the target RFID tag. In the real space, this state means that the object and the user have approached each other. When the reader keeps receiving the signal for some time, the KEEP event is entered, and then the HAS\_A state.

HAS\_A state is the state in which the RFID reader continuously receives the signal from the target RFID tag. In the real space, this state means that the user is carrying the target object. When the signal from the tag is broken (LOST event), the WAITS\_FOR state is entered.

### 3.2.3 Event Cache Construction

Figure 3 illustrates the construction of the event cache. Each entry in the event cache is composed of an entry identifier part, target RFID part, event detection condition part, reaction definition part, and linked entry definition part. The entry identifier part is the entry identification number in the event cache. The target RFID part describes the RFID value to be observed.

In some kinds of services, such as the shopping list service described in Section 2.1, the user terminal should be in the WAITS\_FOR state where the RFID values are unknown. That is, the user has decided the kind of object to be bought, but does not know the entire RFID value. In order to support such situations, the authors propose to introduce the standardized category field that identifies the kind of the object into the RFID format. This allows the event cache to detect the event by the category field value (a part of the RFID) instead of the entire RFID value. In Figure 3, entry #02 WAITS\_FOR any object belonging to category “beef (indicated by the code C)”.

The Event detection condition part defines the detection condition of the event according to the tag state condition and the time condition. The tag condition can be specified by not only the state of the observed RFID tag state but also the related RFID tag state. The related tag state is assumed to define the place condition retrieved by the locator tag. The reaction definition part defines the reaction that is to be executed by the user terminal actuator and the execution priority of the reaction when the event detection condition is satisfied. If the communication link is available, when the event occurs, the execution priority provides the definition of whether priority is given to the execution of the reaction (P (A) in Figure 3) or to communication with the server which will then execute advanced inference including the situation around the user (P (B) in Figure 3).

Entry ID	Target RFID		Event detection condition				Linked entry def.	Reaction definition
	tag ID	Category	State of tag	Related tag		Time		
				tag ID	State			
#01	101	-	HAS_A	-	-	-	-	B, M (“You dropped Purse!”), P(A)
#02	-	xCx	WAITS_FOR	303	DETECTED	-	-	L, M (“Please buy beef at super”), P(B)
#03	301	-	DETECTED	-	-	-	#04	L, M (“Please go to the super”), P(B)
#04	-	-	-	-	-	18:30	#03	L, M (“Please go to the super”), P(B)

xCx: x means ignored field, C is category identifier for the object “beef”. B, L, M (), and P () are flags of Beeping, Lighting, Messaging on display, and priority.

Fig. 3. Construction of the event cache

It is considered that there will be cases in which several events are linked, so if one specific event occurs then the other will no longer be possible or required. For this case, if the first event cache entry is hit, the second should be deleted automatically. The linked event part provides support for these linked events. In Figure 3, if entry #03 (#04) hit, entry #04 (#03) is automatically deleted.

### 3.3 Service Scenarios Using Event Cache

In this section, the authors describe an example of event cache activities in the case of the Lost Property Notification service.

The user starts the service through the user terminal when he/she leaves home. The server recognizes the user-carried objects by observing the RFID tags collected by the user terminal over some period. If the user terminal receives a signal from an RFID constantly, the corresponding entry of the event cache moves to the state of "HAS\_A". As a result of this process, entries #01 and #02 in Figure 4 are generated and downloaded to the user terminal. If the server detects the possibility of the user's visiting his/her friend's home from the tracking information from the received locator tag, the server adds entry #03, see Figure 4.

While the user is in a subway car and the mobile communication network cannot be used, if object "Purse" is dropped, event cache #01 is activated and the user terminal issues a maximum strength warning (Beeping, Vibrating, Lighting, and Messaging) to the user at once. On the other hand, if the place where the event occurs is the friend's house and if the object is not a purse but an umbrella, event cache #03 is activated and the user terminal tries to communicate to the server prior to notifying the user, because the execution priority of #03 is B. If the mobile communication network cannot be used, the user terminal issues a warning via the message display. If the user terminal can link to the server, it transfers all logged IDs to the server. The server then retrieves information about these objects. If there are many objects owned by the friend where the event occurred, the server recognizes that the user left it at the friend's home intentionally, and changes the state of the umbrella to WAITS\_FOR.

Entry ID	Target RFID		Event detection condition				Linked entry def.	Reaction definition
	Tag ID	Cat.	State of tag	Related tag		Time		
				tag ID	State			
#01	101	-	HAS_A	-	-	-	-	B, V, L, M ("You dropped Purse!"), P(A)
#02	201	-	HAS_A	-	-	-	-	V, L, M ("Didn't you leave your umbrella?"), P (B)
#03	201	-	HAS_A	302	DETECTED		-	L, M ("You left or mislaid your umbrella at Mr. A's home"), P(B)

B, V, L, M (), and P () are flags of Beeping, Vibrating, Lighting, Messaging on display, and priority. tag101 is the user's purse, 201 is the user's umbrella, and 302 is the locator tag near the friend's home.

**Fig. 4.** Event cache transfer in the Lost Property Notification Service

## 4 Implementation

The authors implemented a prototype. The RFID tag/reader is based on the SPIDER V system [8]. SPIDER V is an active type tag system, so each tag transmits its ID periodically for the reader to capture it. The event cache was implemented by i-appli (DoJa2.1) [9]. Because it is not possible to connect the cellular phone directly to the SPIDER V reader, the authors used a cellular phone emulator on a note PC. The SPIDER V tag reader was connected to the note PC via a serial interface.

Generally speaking, even if the tag is within the reader's range, the tag reader can not receive 100% of the transmitted signal because of signal failures caused by the characteristic of electric wave propagation and collision with other tag signals. Therefore, the user terminal stores the history of several signaling periods, and the KEEP and LOST events are defined by whether the received signal number exceeds some threshold. This means that there is some delay in detecting the corresponding event. Maximum delay value of the LOST event is as follows.

$$(T-m) \times t+t \text{ [seconds]} \tag{1}$$

where,

T is the number of stored periods, m is a threshold number, and t is tag signaling period.

## 5 Evaluations and Consideration

### (1) Service consistency in the disconnected situation

Figure 5 illustrates the basic operation sequence in the Lost Property Notification service. Here,  $t=1$ ,  $T=10$ , and  $m=3$ . First, service start is acquired and the list of carried objects is displayed (step 1). Next, the communication link between note PC and the server was cut (step 2). The tag corresponding to "Purse" was put into a shielded box (step 3), The Lost event was detected nine seconds after step 3, the cache was executed, and the warning was displayed on the emulator screen (step 4). Finally, the communication link was recovered and the new cache was downloaded by the server and the display was updated (step 5, 6). The results confirmed service consistency in the disconnected situation.

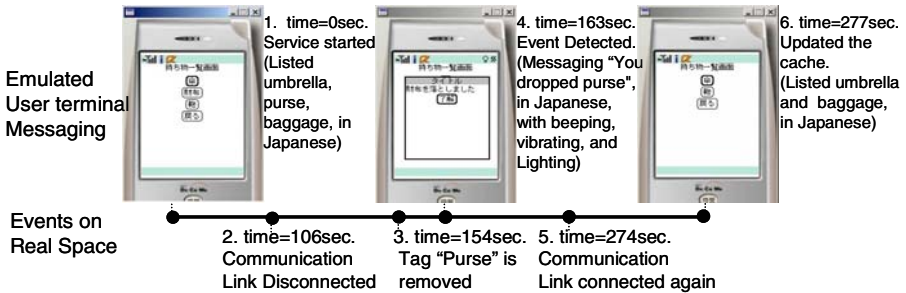


Fig. 5. Basic operation sequence in the Lost Property Notification service

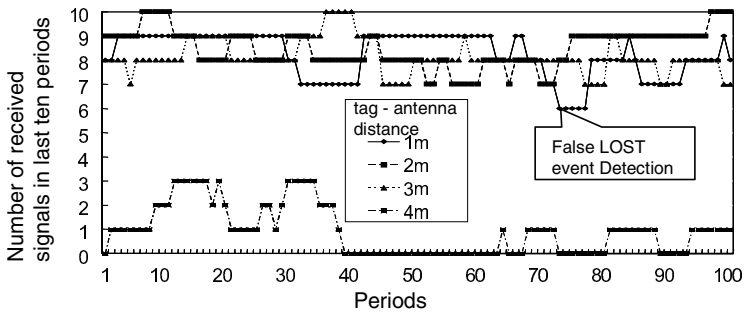
(2)Event detection delay

From equation (1), the event detection delay can be reduced if  $m$  is increased, or  $T$  or  $t$  is lessened. Because excessively small  $t$  causes tag signal collision and shortens the life of the tag battery, the authors fixed  $t$  to 1 and tried to enlarge  $m$  and lessen  $T$ . Table. 1 shows the experimental results (ten times average and standard deviation) and the ideal value from equation (1) of event detection delay when  $T=10/m=3$ ,  $T=10/m=7$ , and  $T=6/m=3$ . The difference, a few seconds, between the ideal value and the experimental value is due to the communication delay of the tag collecting function and event cache, thread switching timing, etc. In addition, when execution priority is B, another few seconds of delay is added for communication to the server.

**Table 1.** Evaluation results of event detection delay

	Average detection delay (standard deviation) [seconds]		
	$T=10/m=3$	$T=10/m=7$	$T=6/m=3$
Ideal value	8	4	4
execution priority A	9.3(0.63)	5.3(0.64)	5.1(0.66)
execution priority B	11.1(0.58)	7.0(0.54)	6.5(0.57)

The results of Table 1 show that actually the delay was lessened when the  $T=10/m=7$  and  $T=6/m=3$ . However, too large  $m$  and too small  $T$  causes another problem. Figure 6 illustrates the 100 signaling period monitoring results of the event cache internal parameter for event detection (i.e. when this value becomes less than  $m$  the event cache detects the LOST event) when  $m$  is set to 7. The Y axis shows how many signals from the tag were received in the last  $T$  (in this case  $T=10$ ) periods, and the X axis shows the number of periods. Figure 6 shows that the false event of LOST was detected even though the tag - antenna distance was 1 meter. This is considered to reflect the influence of signal failure described in Section 4. Similarly, too small  $T$  values ( $T=6/m=3$ ) caused the false detection as in the other experiment. The user will not adopt the ubiquitous service if there are many annoyances like false detection. On the contrary, too large  $T$  and too small  $m$  enlarge the delay. The authors think that  $m$



**Fig. 6.** Tag detection history: ten cycle windows over 100 cycles

should be 3 and T should be set to 10 according to the results of Figure 6. The average delay is 9.3 seconds from Table 1. The authors consider that this result satisfies the real-time requirements for mobile user observation ubiquitous services described in Section 2.1.

## 6 Conclusion

In this paper the authors examined the user terminal needed for realizing ubiquitous computing services. Our terminal is based on the thin client architecture and uses event caching in order to provide service continuity even if the communication link between the client and the server is disconnected. A prototype and evaluation results were described. The results indicate that the proposal is sufficiently practical. Detailed evaluations such as event prediction and field experiments are being planned.

## References

1. Weiser, M.: The Computer for the 21st Century. *Scientific American* (1991) 415-438
2. Takahashi, T., Mizuno, T.: Thin Client-based Handheld Device Architecture for Ubiquitous Computing. *Proceedings of Workshop on Informatics 2004* (2004) 330-334, in Japanese
3. MIT PROJECT OXYGEN. <http://oxygen.lcs.mit.edu/>
4. Ubiquitous ID Center. <http://www.uidcenter.org>
5. Sinclir, J., Merkow, M.: *Thin Clients Clearly Explained*. Morgan Kaufmann, San Francisco (2000)
6. Kanter, J.: *Thin Clients/Server Computing*. Microsoft Press, Washington (1998)
7. Takahashi, T., Takahashi, O., Mizuno, T.: A Study of a Thin Client System for Mobile Computing. *IPSJ Journal*, Vol.45, No.5 (2004) 1417-1431, in Japanese
8. SPIDER V System. <http://www.nextcom.co.jp/solutions/rfidrfid/spiderv.htm>
9. [http://www.nttdocomo.co.jp/p\\_s/imode/make/java/index.html](http://www.nttdocomo.co.jp/p_s/imode/make/java/index.html)



# An Application Development Environment for Rule-Based I/O Control Devices

Ryohei Sagara<sup>1</sup>, Yasue Kishino<sup>1</sup>, Tsutomu Terada<sup>1</sup>, Tomoki Yoshihisa<sup>2</sup>,  
Masahiko Tsukamoto<sup>3</sup>, and Shojiro Nishio<sup>1</sup>

<sup>1</sup> Graduate School of Information Science and Technology, Osaka University, Japan

<sup>2</sup> Academic Center for Computing and Media Studies, Kyoto University, Japan

<sup>3</sup> Faculty of Engineering, Kobe University, Japan

**Abstract.** In this paper, we propose an application development environment for the ubiquitous chip, which is a rule-based event-driven input/output (I/O) control device for constructing ubiquitous computing environments. The proposed development environment simulates the behaviors of multiple ubiquitous chips and helps users to create rules. Moreover, it has a function for developing applications by cooperation between virtual ubiquitous chips and real ubiquitous chips. The application environment enables both programmers and general users to develop and customize applications for ubiquitous computing environments.

## 1 Introduction

Recent evolutions in the miniaturization of computers and component devices such as microchips, sensors, and wireless modules, contribute to the achievement of ubiquitous computing environments [4, 8, 10]. In our ubiquitous computing environments, small devices are embedded in many places to support daily human life. To construct ubiquitous computing environments, we propose a rule-based I/O control device called *ubiquitous chip* [9].

The behaviors of a ubiquitous chip are described by a set of event-driven rules, and a ubiquitous chip can dynamically change its behavior by modifying stored rules. In our assumed environments, ubiquitous chips are embedded into almost any artifacts to enrich our daily-life, and we can customize functions and services in ubiquitous chips according to our preference.

To achieve such environments, we need an application development environment that enables both programmers and general users to intuitively develop/customize applications. In response to these requirements, we propose a development environment for ubiquitous chips that simulates the behaviors of multiple ubiquitous chips and helps users to create rules. Moreover, this proposed environment includes a function for developing applications through cooperation between virtual and real ubiquitous chips.

The remainder of this paper is organized as follows. Section 2 outlines the ubiquitous chip. Section 3 describes the design of the proposed application development environment, and Section 4 describes a prototype system. Section 5 discusses the development environment and Section 6 sets forth our conclusions and planned future work.

## 2 Ubiquitous Chip

As shown in Figure 1, a ubiquitous chip consists of a core part, which is the main unit, and a cloth part that has connectors and a rechargeable battery. It has five digital input ports, one analog input port, twelve digital output ports, two serial communication ports, and a multi-purpose LED. Figure 2 shows the various input/output devices for the ubiquitous chip such as sensors, input devices, and actuators. Using these attachments, we can flexibly change configurations of ubiquitous chips. The behaviors of ubiquitous chip are described by a set of ECA rules, which are used for describing behaviors in event-driven databases. An ECA rule consists of Event, Condition, and Action. Event is an occurring event, Condition is a condition for executing actions, and Action is the operations to be carried out. Tables 1, 2, and 3 show the lists of events, conditions, and actions that can be used on the ubiquitous chip.

A ubiquitous chip communicates with other ubiquitous chips via its serial communication ports. We can use the SEND\_MESSAGE action, the SEND\_DATA action, and SEND\_COMMAND action as communication functions. The SEND\_MESSAGE action sends a message that has a specific ID (0-7). The SEND\_DATA action sends one byte data that is specified in the rule or input voltage of the analog port. The SEND\_COMMAND action sends a command to remotely manage ECA rules stored in ubiquitous chips. Table 4 shows the lists of commands that can be sent by the SEND\_COMMAND action. The DEMAND\_DATA command demands the one byte data specified address of the memory in a ubiquitous chip. When a ubiquitous chip receives a DEMAND\_DATA command, it returns the required data as a REPLY\_DATA command.

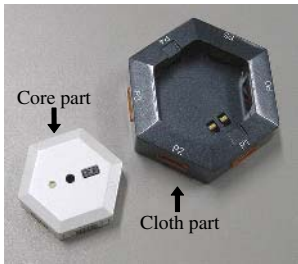


Fig. 1. Ubiquitous chip

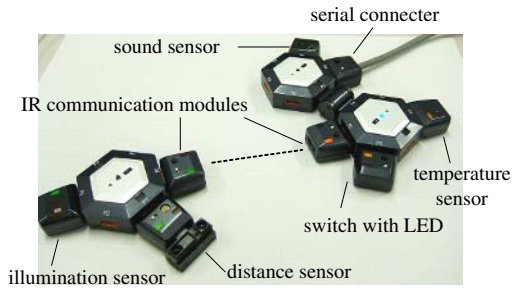


Fig. 2. Attachments for ubiquitous chip

Table 1. Events

Name	Contents
TIMER_EXPIRE	Firing a timer
RECEIVE_MESSAGE	8 types of message reception via a serial port
RECEIVE_DATA	1 byte data reception via a serial port
NONE	Evaluating conditions at all times

**Table 2.** Conditions

Name	Contents
INPUT	On/Off state of input ports
ANALOG_INPUT	Range of input from the analog port
INPUT_STATE	Value of internal variables
TIMER_ID	ID of fired timer
MESSAGE_ID	ID of received message
DATA_RANGE	Range of received data

**Table 3.** Actions

Name	Contents
OUTPUT	On/Off control of output ports
OUTPUT_STATE	On/Off control of state variables
TIMER	Setting a new timer
SEND_MESSAGE	Sending a message
SEND_DATA	Sending a 1 byte data
SEND_COMMAND	Sending a command
HW_CONTROL	Hardware control

**Table 4.** Commands

Name	Contents
ADD_ECA	Adding a new ECA rule
DELETE_ECA	Deleting a specific ECA rule(s)
ENABLE_ECA	Enabling a specific ECA rule(s)
DISABLE_ECA	Disabling a specific ECA rule(s)
DEMAND_DATA	Requesting a data of EEPROM
REPLY_DATA	Sending a data of EEPROM (reply to DEMAND_DATA)

### 3 Design of Application Development Environment

#### 3.1 Requirements

In this research, we assume that ubiquitous chips are embedded into almost any artifacts such as furniture, appliances, walls, and floors, and that they cooperate with each other and provide various services. These services are required to be adaptable to user preferences, as users may want to customize services according to their own requirements. For example, we envisage the following situations:

- When a user buys a new piece of furniture that features an embedded ubiquitous chip and sensors, he/she customizes a room automation application, which is already available in his/her room to integrate the new furniture into the application.
- When a user redecorates his/her room, he/she modifies the application according to the new allocation.
- When a user changes his/her routine, he/she adjusts the applications.
- A user uses actual I/O devices to check the behavior of an application.

We construct an application development environment for ubiquitous chips that visualizes the behavior of applications and achieves easy development for users.

Moreover, the development environment also provides a function for verifying applications with actual I/O devices and ubiquitous chips to enable users to develop/customize applications intuitively.

### 3.2 Approach

In order to satisfy the above requirements, our application development environment has the following functions.

#### Simulation with Virtual Ubiquitous Chips

Services in ubiquitous computing environments are realized through cooperation among multiple ubiquitous chips. In such situations, it is difficult for users to grasp the existing configurations and construct applications taking into consideration of the relationships among multiple ubiquitous chips. Therefore, our application development environment needs a function that simulates multiple virtual ubiquitous chips, which process their ECA rules in the same way as the real ubiquitous chip. A virtual ubiquitous chip has the following characteristics:

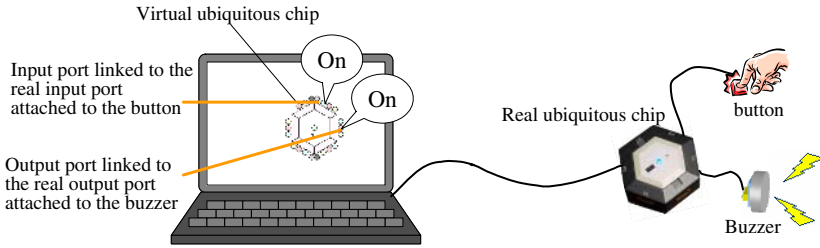
- A virtual ubiquitous chip has a hexagonal shape, I/O ports, serial ports, and a multi-purpose LED, the same as a real ubiquitous chip.
- An arbitrary number of virtual ubiquitous chips can be added/deleted to/from the simulation environment. A user can add/delete connections between I/O ports and serial ports over multiple ubiquitous chips.
- The state of I/O ports and the multi-purpose LED are displayed at all times as a series of colored circles.
- A user can check a virtual ubiquitous chip's internal variables and stored ECA rules even when an application is running.
- A user can add new ECA rules easily without professional knowledge. The application development environment has an ECA rule editor, which enables general users to write ECA rules easily. Moreover, a user can check stored ECA rules in a style similar to natural language.

#### Cooperation Among Real/Virtual Ubiquitous Chips

The application development environment has a function for constructing applications through cooperation between a virtual ubiquitous chip and a real ubiquitous chip. This function achieves the following implementation styles:

- Case 1.** A user customizes the application that is in-service on a real ubiquitous chip.
- Case 2.** A user checks the behaviors of real I/O devices at the final step of application development.

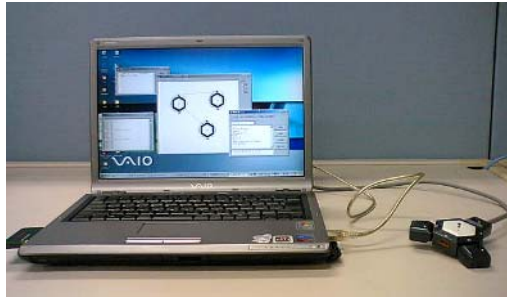
The application development environment manages the state of a real ubiquitous chip in the same way as a virtual ubiquitous chip by linking their states. For example, as Figure 3 shows, when a user pushes the button connected to the real ubiquitous chip, the input port of the associated virtual ubiquitous chip is turned on. Likewise, when the output port of the virtual ubiquitous chip is turned on, the output port of the real ubiquitous chip is also turned on.



**Fig. 3.** Cooperation among real and virtual ubiquitous chips

## 4 Implementation

We have implemented a prototype of the application development environment. In this section, we explain the details of its implementation and show an example of its use. Figure 4 shows a snapshot of the application development using a PC and a real ubiquitous chip.



**Fig. 4.** Example of a application development using a PC and a ubiquitous chip

### 4.1 Simulation with Virtual Ubiquitous Chips

Figure 5 shows a screenshot of the development environment. In the proposed development environment, the behaviors of ubiquitous chips are simulated by virtual ubiquitous chips. A virtual ubiquitous chip is illustrated as a hexagon and the circles indicate I/O ports, serial ports, and a multi-purpose LED. One input port and two output ports are placed along each edge of the hexagon, likewise in the real ubiquitous chip. The state of the I/O ports and the multi-purpose LED are expressed by differences in their color. A user operates the virtual ubiquitous chip in the following ways:

- places multiple virtual ubiquitous chips in the simulation area.
- toggles input ports.
- checks the state of the output ports and the multi-purpose LED.

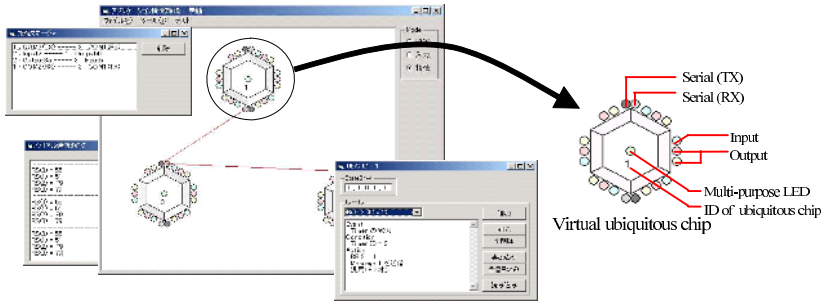


Fig. 5. Screenshot of the application development environment

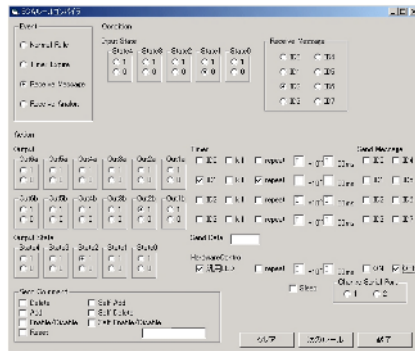


Fig. 6. ECA rule editor

- connects I/O ports and serial ports to the other ubiquitous chips.
- checks the value of the internal variables and the stored ECA rules.
- adds new ECA rules using ECA rule editor (Figure 6).

Users have only to use a mouse to achieve the above operations.

#### 4.2 Cooperation Among Real/Virtual Ubiquitous Chips

As described in Section 3.2, cooperation is classified into two cases: a user customizes an application using a real ubiquitous chip and a user checks the behaviors of real I/O devices.

In the former case, cooperation is achieved as follows:

- Step 1.** A user connects a real ubiquitous chip to the PC.
- Step 2.** A user places a new virtual ubiquitous chip in the simulation area.
- Step 3.** The application development environment reads the ECA rules stored in the real ubiquitous chip and adds them to the virtual ubiquitous chip.
- Step 4.** The development environment sends the DELETE\_ECA command to the real ubiquitous chip to delete all stored ECA rules, this prevents conflict among the ECA rules.

**Table 5.** Formula for creating control rules

Original rule	Control rule
E: (Any event) C: $I(i)=0$ ( $i=1-5$ ) A: (Any action)	E: NONE C: $I(i)=0$ , $S(i-1)=1$ A: $S(i-1)=0$ , SEND_DATA( $2i-1$ )
E: (Any event) C: $I(i)=1$ ( $i=1-5$ ) A: (Any action)	E: NONE C: $I(i)=1$ , $S(i-1)=0$ A: $S(i-1)=1$ , SEND_DATA( $2i$ )
E: (Any event) C: (Any condition) A: $O(i)=0$ ( $i=1-12$ )	E: RECEIVE_DATA C: RECEIVED_DATA= $2i-1$ A: $O(i)=0$
E: (Any event) C: (Any condition) A: $O(i)=1$ ( $i=1-12$ )	E: RECEIVE_DATA C: RECEIVED_DATA= $2i$ A: $O(i)=1$
E: (Any event) C: (Any condition) A: HW_CONTROL	E: RECEIVE_DATA C: RECEIVED_DATA= $25$ A: HW_CONTROL
E: (Any event) C: (Any condition) A: HW_CONTROL(M_LED OFF)	E: RECEIVE_DATA C: RECEIVED_DATA= $26$ A: HW_CONTROL(M_LED OFF)

**Step 5.** The development environment writes rules to the real ubiquitous chip, which lets the real ubiquitous chip behave in the same manner as the virtual ubiquitous chip.

In the latter case, cooperation is realized by performing only Steps 4 and 5 of the above procedure.

Table 5 shows the formula for creating control rules. When the state of a real input port changes, the real ubiquitous chip sends one byte data to the development environment. When the development environment receives the data, it changes the state of the associated virtual input port.

### 4.3 Example

In this section, we give an example of the use of the proposed application development environment. The sample application behaves as though “a user is sitting on a chair, and the desk lamp lights automatically when there is not enough bright.” In this application, we use three ubiquitous chips called UC1, UC2, and UC3. UC1 is attached to the chair and has a pressure sensor that detects when the user is sitting. UC2 is attached to the desk and is connected to the desk lamp in order to control it. UC3 is attached to the wall and has an illumination sensor. Figure 7 shows the connection relationship of the ubiquitous chips.

The user programs the application in the following way:

1. The user positions the three virtual ubiquitous chips, UC1, UC2, and UC3.
2. The user connects the I/O ports and serial ports as shown in Figure 7.
3. The user adds the ECA rules shown in Table 6 using the ECA rule editor.

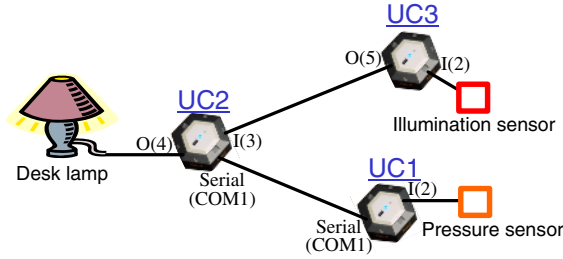


Fig. 7. System structure of sample application

Table 6. Rule set for the sample application

Rules for UC1 (2 rules)			
E: NONE	E: NONE		
C: I(2)=1, S(0)=0	C: I(2)=0, S(0)=1		
A: S(0)=1, SEND_MESSAGE(M0)	A: S(0)=0, SEND_MESSAGE(M1)		
Rules for UC2 (5 rules)			
E: RECEIVE_MESSAGE	E: RECEIVE_MESSAGE	E: NONE	
C: MESSAGE_ID=0	C: MESSAGE_ID=1	C: I(3)=1	
A: S(0)=1	A: S(0)=0, O(4)=0	A: S(1)=0, O(4)=0	
E: NONE	E: NONE		
C: I(3)=0	C: S(0)=1, S(1)=1		
A: S(1)=1	A: O(4)=1		
Rules for UC3 (2 rules)		Control rules for UC3 (2 rules)	
E: NONE	E: NONE	E: NONE	E: NONE
C: I(2)=1	C: I(2)=0	C: I(2)=0, S(2)=1	C: I(2)=1, S(2)=0
A: O(5)=1	A: O(5)=0	A: S(2)=0, SEND_DATA(3)	A: S(2)=1, SEND_DATA(4)

- The user checks the behavior of the ubiquitous chips by toggling their input ports. If bugs are found, the user modifies the rules.
- When the user wants to confirm the behavior of the application with a real illumination sensor, he connects a real ubiquitous chip to a PC and links UC3 and the real ubiquitous chip. In this case, the control rules shown in Table 6 are automatically added to the real ubiquitous chip. The user changes the brightness of the room and checks the behavior.
- When he completes the application, he writes ECA rules to real ubiquitous chips and attaches them to furniture.

## 5 Consideration

### 5.1 Planned Functions

When more than seven virtual ubiquitous chips are placed, the simulation area becomes full. Thus, it is difficult to develop applications consisting of ten or more



ubiquitous chips. To solve this problem, we are planning to develop functions that can group several ubiquitous chips into a meaningful unit and that can manage groups collectively.

Although we can grasp the state of I/O ports through the circles of a virtual ubiquitous chip, we cannot know the behavior of connected devices. Therefore, we should provide virtual I/O devices and functions that can simulate their behaviors.

In this paper, we focus on serial ports connected by means of wired cables. Practically, we have provided various wireless communication units for ubiquitous chips such as Infrared (IR) units, Radio Frequency (RF) units, and Bluetooth units. The development environment should be able to support to simulate wireless communication.

## 5.2 Related Work

Smart-It [2], MOTE [3], and U-Cube [5] are small devices for constructing ubiquitous computing environments and sensor networks. These devices have sensors/actuators and wireless modules and they are similar to ubiquitous chip in the point that we can customize system configurations by changing the attached devices. However, we cannot change their behaviors or the attached devices while applications are running. Therefore, it is difficult to dynamically customize the behaviors of embedded devices according to user demands. Moreover, since these devices are developed with a C-like programming language, it is difficult for general users to develop and customize applications.

MINDSTORMS [6] and ROBOT WORKS [1] have application development environments for specific hardware. Users can easily program applications by aligning blocks in which conditions and operations are described. However, these development environments do not have simulation functions. Moreover, they cannot develop applications through cooperation with actual hardwares.

MPLAB [7] is a development environment for PIC, which is a microprocessor used in ubiquitous chip. MPLAB can simulate the behaviors of PIC by displaying the values of variables. However, it cannot simulate the behaviors of multiple PICs and it cannot visualize the states of I/O ports.

## 6 Conclusion

In this paper, we described the design and implementation of an application development environment for ubiquitous chips. The proposed development environment simulates the behaviors of multiple ubiquitous chips. Moreover, it has a function for verifying applications through cooperating with real ubiquitous chips.

In future, we have plans to construct functions for developing large-scale applications, for simulating I/O devices, and for cooperating with multiple real ubiquitous chips. We also plan operational tests and further evaluation of the application development environment.

## Acknowledgement

This research was partially supported by The 21st Century Center of Excellence Program “New Information Technologies for Building a Networked Symbiotic Environment” and Grant-in-Aid for Scientific Research (A)(17200006) from the Ministry of Education, Culture, Sports, Science and Technology of Japan.

## References

1. BANDAI: “ROBOT WORKS,” <http://www.roboken.channel.or.jp/borg/>.
2. M. Beigl and H. Gellersen: “Smart-Its: An Embedded Platform for Smart Objects,” *Smart Objects Conference (sOc)* (May. 2003).
3. Crossbow Technology Inc.: “MICA,” [http://www.xbow.com/products/Wireless\\_Sensor\\_Networks.htm](http://www.xbow.com/products/Wireless_Sensor_Networks.htm).
4. J. Kahn, R. Katz, and K. Pister: “Mobile Networking for Smart Dust,” in *Proc. ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom99)*, pp. 271–278 (Aug. 1999).
5. Y. Kawahara, M. Minami, H. Morikawa, and T. Aoyama: “Design and Implementation of a Sensor Network Node for Ubiquitous Computing Environment,” in *Proc. VTC2003-Fall* (Oct. 2003).
6. LEGO: “MINDSTORMS,” <http://mindstorms.lego.com/japan/products/>.
7. Microchip Technology Inc.: “MPLAB,” <http://www.microchip.com/1010/index.htm>.
8. K. Sakamura: “TRON: Total Architecture,” in *Proc. Architecture Workshop in Japan'84*, pp.41–50 (Aug. 1984).
9. T. Terada, M. Tsukamoto, K. Hayakawa, T. Yoshihisa, Y. Kishino, S. Nishio, and A. Kashitani: “Ubiquitous Chip: a Rule-based I/O Control Device for Ubiquitous Computing,” in *Proc. Int'l Conf. on Pervasive Computing (Pervasive 2004)*, pp.238–253 (Apr. 2004).
10. M. Weiser: “The Computer for the Twenty-first Century,” *Scientific American*, Vol. 265, No. 3, pp. 94–104 (Sept. 1991).

# A uWDL Handler for Context-Aware Workflow Services in Ubiquitous Computing Environments

Yongyun Cho, Joohyun Han, Jaeyoung Choi, and Chae-Woo Yoo

School of Computing, Soongsil University,  
1-1 Sangdo-dong, Dongjak-gu, Seoul 156-743, Korea  
{jhhan, yycho}@ss.ssu.ac.kr, {choi, cwwoo}@comp.ssu.ac.kr

**Abstract.** To develop context-aware workflow services in ubiquitous computing environments, a service developer must describe and recognize context information as transition constraints. uWDL (ubiquitous Workflow Description Language)[1] is a workflow language that describes the situation information of ubiquitous environments as a rule-based service transition condition. In this paper, we suggest a uWDL handler that supports workflow's service transition to be aware of user's condition information. The uWDL handler consists of a uWDL parser and a uWDL context mapper. The uWDL parser represents contexts described in the scenario with sub-trees of a DIAST (Document Instance Abstract Syntax Tree) as a result of the parsing. To derive the right transition of workflow services, the uWDL context mapper compares contexts described in sub-trees of DIAST with a user's situation information generated from ubiquitous environments by using a context comparison algorithm. Therefore, the uWDL handler will be used in developing context-aware workflow applications that can change the flow of a service scenario according to the user's situation information in the ubiquitous computing environment.

## 1 Introduction

Ubiquitous computing environments mean that a user can connect with a network freely and receive services that he wants, anyplace and anytime [2, ?]. A workflow model for business services in traditional distributed computing environments [3] can be applied as a service model to connect services related in ubiquitous computing environments and express service flows [1]. However, a workflow in ubiquitous computing environments must decide a service transition according to the user's situation information that is inputted dynamically [2]. For that, a workflow language for ubiquitous environments must be able to express the user's situation information as service transition conditions in a workflow service scenario. uWDL (ubiquitous Workflow Definition Language) is a workflow language based on a structural context model which expresses context information as transition constraints of workflow services [1, ?]. Through a workflow service scenario document in uWDL, developers can represent context information as workflow state transition constraints in order to support a context-aware service transition of workflows. To develop application programs

with a workflow language, a developer commonly needs a handler that processes a document written in that language and interprets the structure and the meaning of it.

In this paper, we present a uWDL handler that verifies the validation of a uWDL workflow service scenario document and derives the service transition according to a user's state information being inputted dynamically in ubiquitous environments. For that, the uWDL handler consists of a uWDL mapper and a uWDL parser. The uWDL parser parses a uWDL scenario document, and produces DIAST (Document Instance Abstract Syntax Tree), which represents the document's structure information. To decide a workflow service transition, a uWDL mapper compares contexts described in DIAST with the user's situation information offered from a sensor network.

## 2 Related Work

### 2.1 Context-Aware Workflow and Workflow Language

Context in a ubiquitous environment means any information that can be used to characterize the situation of an entity [3]. An application or system that uses context information or performs context-appropriate operations is called a context-aware application or context-aware system [4, ?]. Ubiquitous workflow is dependent on context information that is sensed from the physical environment, and provides a context-aware service automatically based on that sensed information. The ubiquitous workflow is required to specify ubiquitous context information as state-transition constraints. The existing workflow languages, such as BPEL4WS [5], WSFL [6], and XLANG [7], are suitable for business and distributed computing environments. These languages use the results of the former services and the event information of services as transition conditions of services. However, they do not include any elements to describe context in ubiquitous computing environments to workflow services. For example, XPath is unsuitable for expressing high-level situation information that comes from ubiquitous environments, because it has only logic and condition operators.

### 2.2 uWDL (Ubiquitous Workflow Description Language)

uWDL [1] can describe context information as transition conditions of services through the <context> element consisting of the knowledge-based triple entity - subject, verb, and object. The uWDL reflects the advantages of current workflow languages such as BPEL4WS, WSFL, and XLANG, and also contains rule-based expressions to interface with the DAML+OIL [8] ontology language. In uWDL, a simple context and profile information are described using an RDF expression [9], and complex context information is expressed using an ontology expression. Figure 1 shows uWDL's schema.

In Figure 1, the <node> element points to Web services in ubiquitous environments and it conforms to a web service's operation. The <link> element contains

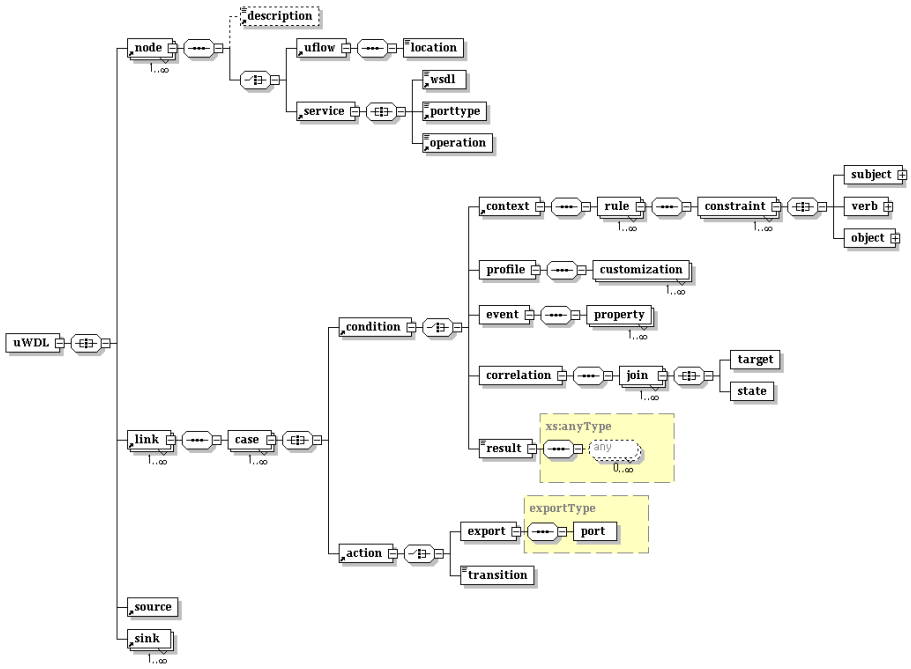


Fig. 1. uWDL Schema

a sub element <condition>, which selects the service flows. The <context> element contains the <constraint> element in order to specify high-level context information generated by ontology and inference services as a form of structural description. The <constraint> element has the triplet sub-element of <subject>, <verb>, and <object> based in RDF. The <node> element points to one operation that provides a functionality of Web services in ubiquitous environments. The <transition> element specifies the state change of a current node. The <condition> element makes a decision to select a proper service by context, profile, and event information. The composite attribute of the <constraint> element has a value of 'and', 'or', and 'not'. By using these attributes, we can express the relationship between simple contexts and describe a high-level complex context. The <rule> element means a set of the <constraint> elements.

### 2.3 Context Parser

To develop an application program in a context definition language, a developer needs a parser to parse the structure and the meaning of a document that is made out in the language. Jena2 [11] is useful as a parser for such ontology languages as RDF, DAML+OIL, and OWL [12] to define context. Jena2 parses ontology based in existent RDF as well as contexts of DAML+OIL, OWL, N3, DB, and so on. Jena2 redefines low-level contexts from a sensor network in

ubiquitous environments as high-level contexts based in RDF. However, Jena2 is not suitable for processing workflow scenario documents like a uWDL scenario document that includes the <context> element to describe contexts. Therefore, developers may require a parser that can recognize and parse workflow service scenario documents describing contexts like a uWDL workflow service scenario document in ubiquitous computing environments.

In this paper, we design and implement a uWDL parser that parses the structure and meaning of a uWDL document. Also, we propose an algorithm for comparing contexts inputted in RDF-based triple form in ubiquitous environments with contexts described in uWDL workflow service scenario documents. Through the algorithm, the uWDL mapper makes the workflow perform context-aware service transitions according to a user's situation.

### 3 A uWDL Handler

#### 3.1 A System Architecture

In this paper, we propose a uWDL handler that can help a service developer to develop context-aware workflow services in ubiquitous computing environments. Figure 2 shows the system structure of the uWDL handler.

After a service developer writes a uWDL workflow service scenario, the uWDL handler compares a context described as subject, verb, and object elements in the uWDL scenario to other contexts, which are obtained as the entity from a sensor network. To do that, we suggest an algorithm to parse uWDL service scenarios and recognize the context according to the sensed contexts from the

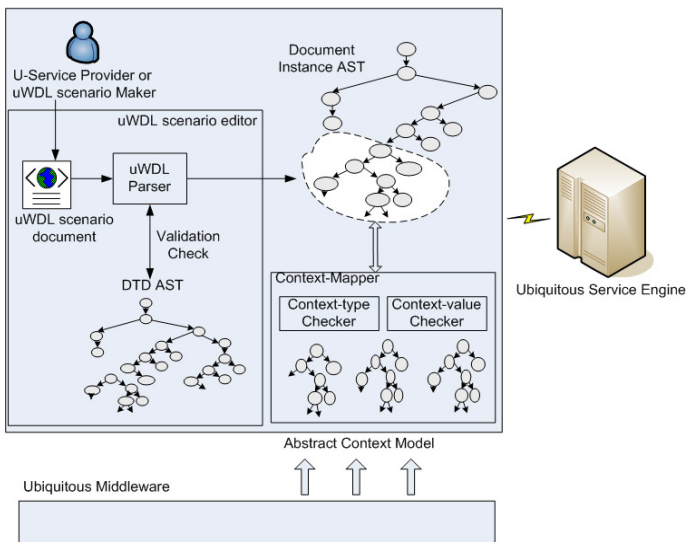


Fig. 2. The architecture for handling the context in uWDL



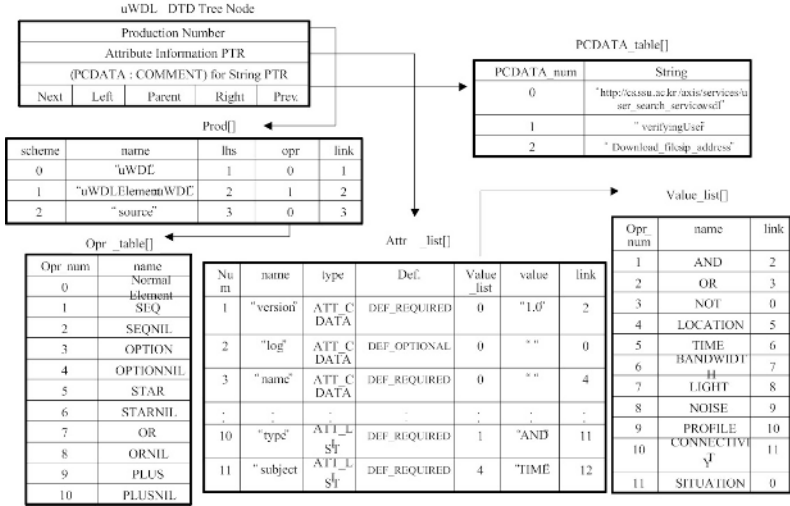


Fig. 4. DIAST’s element node structure

represented as RDF-based triple nodes. Therefore, a <constraint>’s subtree in the DIAST is compared with a user’s situation information inputted from sensors to derive service transitions described in a workflow scenario. Figure 4 displays the data structure of the DIAST’s element node and various data tables. The DIAST’ nodes are connected to each other and useful data tables by 6 pointers.

Through the data structure of the element node and its data tables, developers can easily know the whole DIAST structure and use it to compare contexts between DIAST and sensors in ubiquitous computing environments. The Production Number is a unique element number which distinguishes each element node. The Left, Parent, and Right links express for element’s order and connection information in the DIAST. Each node in the DIAST is divided into a common element node or an operator node. An operator node displays a meta-character to express a language-specific characteristic of elements in a uWDL’s DTD. For example, a parent element node for <node> element in the example DIAST of Figure 3 is <PLUS> operator node, not a common element node. The Attribute Information PTR is a pointer that indicates a relevant record of Attr\_list [] to get an element’s attribute value. The String PTR is a pointer that indicates a record of the PCDATA\_table that contains string information of PCDATA or COMMENT element.

### 3.3 uWDL Context Mapper and Context Comparison Algorithm

Contexts that the context mapper uses for the comparison are described in a triple entity based on RDF. Context information from the sensor network can be embodied as a triple entity consisting of subject, verb and object according to the structural context model based in RDF. A context described in the



---

```

Boolean MatchContext(UC A, OCS B) {
  int j; /* For the index of context in B each context set */
  for each j in OCS B { /* Repeatedly comparing contexts in A, B context set */
    if ((A.UCs_type == Bj.OCS_type && A.UCs_value == Bj.OCS_value) &&
        (A.UCv_type == Bj.OCv_type && A.UCv_value == Bj.OCv_value) &&
        (A.UCo_type == Bj.OCo_type && A.UCo_value == Bj.OCo_value))
      return TRUE /* Found context match */
    } /* End for */
  return FALSE; /* Return matchresult */
}

```

---

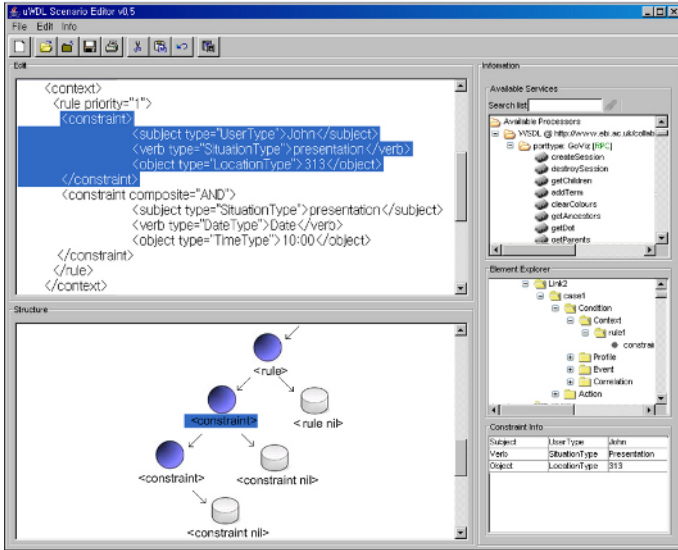
**Fig. 5.** An algorithm for comparing UC A with OCS B

<constraint> element in the uWDL service scenario consists of the triple entity based in RDF. The context mapper extracts context types and values of the entity objectified from sensors. It then compares the context types and values of the objectified entity with those of the DIAST's subtree elements related to the entity. In the comparison, if the context types and values in the entity coincide with the counterpart in the DIAST's subtree, the context mapper drives the service workflow. A context comparison algorithm is shown in Figure 5.

In Figure 5, we define a context embodied with a structural context model from the sensor network as  $OC = (OCs\_type, OCs\_value), (OCv\_type, OCv\_value), (OCo\_type, OCo\_value)$ , and a context described in a uWDL scenario as  $UC = (UCs\_type, UCs\_value) (UCv\_type, UCv\_value), (UCo\_type, UCo\_value)$ . OC means a context objectified with the structural context model, and it consists of OCs, OCv, and OCo which mean subject, verb, and object entities. UC means a context described in a uWDL. UCs, UCv, and UCo mean subject, verb, object entities in the uWDL scenario. A context consists of a pair of type and value. Also, OCS and UCS that mean each set of OC and UC can be defined as  $OCS = (OC1, OC2, OC3, \cdot, OCi)$  and  $UCS = (UC1, UC2, UC3, \cdot, UCi)$ .

## 4 Experiments and Results

For testing, we will make a uWDL scenario for an office meeting service in ubiquitous environments, and show how the suggested uWDL handler makes the workflow's service perform context-aware transitions, by comparing contexts described in the scenario with a user's situation information from sensors. The example scenario is as follows. John has a plan to do a presentation in Room 313 at 10:00 AM. When John moves to Room 313 to participate in the meeting before 10:00 AM, a RFID sensor above room 313's door transmits John's basic context information (such as name, notebook's IP address) to a server. If the conditions, such as user location, situation, and current time, are satisfied with contexts described in the uWDL workflow service scenario, then the server downloads his presentation file and executes a presentation program. A developer can use <subject>, <verb> and <object> in the uWDL scenario to decide what

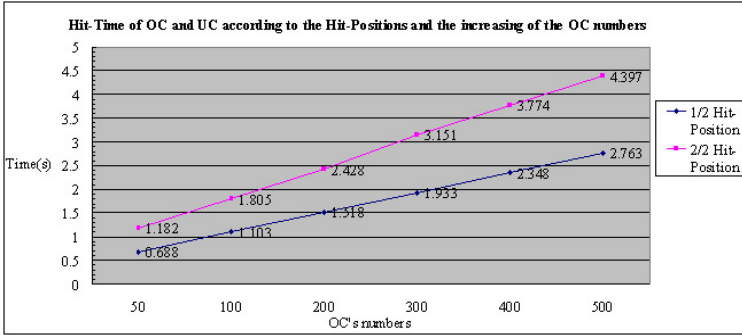


**Fig. 6.** The service scenario and the DIAST’s subtree that the uWDL parser in an uWDL scenario editor produced for the scenario

service is selected, according to context and profile that are the user’s situation information.

For experiments, we implement a uWDL scenario editor. The editor includes the suggested uWDL handler and gives convenient user interfaces to set constraints’ values of a context. Figure 6 shows a uWDL workflow service scenario for the example scenario, and a part of the <constraint> subtree of DIAST that the WDL parser produces for the uWDL scenario. The subtree in the structure window is for the <constraint> highlighted in the edit window. Now, if the context mapper receives context data objectified as (SituationType, presentation), (UserType, Michael), (UserType, John), and (LocationType, 313), it compares the contexts’ types and values with the subtree’s elements shown in Figure 6. In this case, because context (UserType, Michael) is not suitable anywhere in the subtree’s elements, it is removed. The context described to the uWDL scenario in Figure 6 consists of a limited number of UCs. However, contexts from a sensor network can be produced as innumerable OCs according to the user’s situation. Therefore, the uWDL handler must quickly and correctly select an OC coinciding with a UC that is described in the uWDL scenario from such innumerable OCs. In an experiment, we generated a lot of OCs incrementally, and measured how fast the suggested uWDL handler found the OCs that coincided with the UCs in the uWDL scenario of Figure 6. We used a Pentium 4 2.0 Ghz computer with 512M memory based in Windows XP OS for the experiment. Figure 7 is the result.

In Figure 7, we increased the OC’s amounts by 50, 100, 200, 300, 400 and 500 incrementally. We placed the OCs coinciding with the UCs in the middle and end of the OCs that we produced randomly. In Figure 7, 1/2 hit-position means



**Fig. 7.** A hit-time for hit-position and the number of OCs

the position of the OC coinciding with the UC is the middle of the produced OCs, and 2/2 hit-position means the position of the OC is the end of the OCs. As shown in the result, the hit-time did not increase greatly regardless of the OCS's considerable increase. Also, we verified that all schedule.ppt files had been downloaded in all cases when hits between OCs with UCs happened. It shows that the suggested uWDL handler can sufficiently support context-aware workflow service.

## 5 Conclusion

uWDL is a ubiquitous workflow description language to specify service flows, where appropriate services are selected based on context information and executed concurrently or repeatedly, and to specify context-aware state transition. In this paper, we present the uWDL handler that can process a uWDL workflow service scenario document, and can derive service transition according to a user's situation information. Through experiments, we showed processes in which the uWDL handler parsed the created uWDL scenario document and produced a DIAST for the document. We defined a user's status information from the sensor network as OC based on RDF, and a context described in uWDL scenario as UC. We showed an experiment in which the uWDL mapper compared contexts of UCSs and OCSs through a context comparison algorithm, and measured hit-times and service transition accuracy to verify the efficiency of the algorithm. Through the results, we found that the hit-times were reasonable in spite of the OCs' amounts. Therefore, this uWDL handler will contribute greatly to the development of the context-aware application programs in ubiquitous computing environments.

## Acknowledgements

This research is supported by the Ubiquitous Autonomic Computing and Network Project, the Ministry of Information and Communication (MIC) 21st Century Frontier R&D Program in Korea.

## References

1. Joohyun Han, Yongyun Cho, Jaeyoung Choi: Context-Aware Workflow Language based on Web Services for Ubiquitous Computing, ICCSA 2005, LNCS 3481, pp. 1008-1017, (2005)
2. M. Weiser: Some Computer Science Issues in Ubiquitous Computing. *Communications of the ACM*, Vol.36, No.7 (1993) 75-84
3. D. Hollingsworth: The Workflow Reference Model. Technical Report TC00-1003, Work flow Management Coalition (1994)
4. Guanling Chen, David Kotz: A Survey of Context-Aware Mobile Computing Research, Technical Report, TR200381, Dartmouth College (2000)
5. Tony Andrews, Francisco Curbera, Yaron Goland: Business Process Execution Language for Web Services. BEA Systems, Microsoft Corp., IBM Corp., Version 1.1 (2003)
6. Frank Leymann: Web Services Flow Language (WSFL 1.0). IBM (2001)
7. Satish Thatte: XLANG Web Services for Business Process Design. Microsoft Corp. (2001)
8. R. Scott Cost, Tim Finin: ITtalks: A Case Study in the Semantic Web and DAML+OIL. University of Maryland, Baltimore County, IEEE (2002) 1094-7167
9. W3C: RDF/XML Syntax Specification, W3C Recommendation (2004)
10. James Snell: Implementing web services with the WSTK 3.2, Part 1, IBM Tutorials, IBM (2002)
11. Jena2-A Semantic Web Framework.  
Available at <http://www.hpl.hp.com/semweb/jena1.html>
12. Deborah L. McGuinness, Frank van Harmelen (eds.): OWL Web Ontology Language Overview, W3C Recommendation (2004)
13. Aho, A.V., Sethi R., and Ullman J. D., *Compilers: Principles, Techniques and Tools*, Addison-Wesley (1986)
14. Bates, J. and Lavie A., "Recognizing Substring of LR(K) Languages in Linear Time", *ACM TOPLAS*, Vol.16 ,No.3, pp.1051-1077 (1994)
15. Reckers J. and Koorn W., Substring parsing for arbitrary context-free grammars. *ACM SIGPLAN Notices*, 26(5), pp.59-66 (1991)

# SMMART, a Context-Aware Mobile Marketing Application: Experiences and Lessons

Stan Kurkovsky<sup>1</sup>, Vladimir Zanev<sup>2</sup>, and Anatoly Kurkovsky<sup>3</sup>

<sup>1</sup> Computer Science Department,  
Central Connecticut State University,  
1615 Stanley Street, New Britain, CT 06050, USA  
KurkovskySta@ccsu.edu

<sup>2</sup> Department of Computer Science,  
Columbus State University,

4225 University Avenue, Columbus, GA 31906, USA  
Zanev\_Vladimir@colstate.edu

<sup>3</sup> Department of Mathematics, Physics and Computer Science,  
University of the Sciences in Philadelphia,  
600 South Forty-third Street, Philadelphia, PA 19104, USA  
a.kurkov@usip.edu

**Abstract.** A new class of m-commerce applications is emerging due to the unique features of handheld devices, such as mobility, personalization and location-awareness. This paper presents SMMART, a context-aware, adaptive and personalized m-commerce application designed to deliver targeted promotions to the users of mobile devices. SMMART adapts to changing interests of its user by monitoring his or her shopping habits and guarantees the privacy of its users by not transmitting any personally identifiable information to the retailers. We describe our experiences of building and evaluating a fully functional prototype of SMMART implemented for Pocket PCs.

## 1 Introduction

M-commerce is a branch of electronic commerce, in which mobile devices and their network connection medium are used in the process of buying and selling of services, and products. Wireless mobile devices possess unique features: ubiquity (they are affordable and portable), personalization (a device belongs to and can be identified with a single individual), and location awareness (a wireless connection may be used to determine the physical location) [7]. While some existing e-commerce applications are adapted for mobile platforms, the features inherent to mobility and wireless communication medium create a unique class of emerging m-commerce applications striving to achieve the anytime, anywhere paradigm of pervasive computing [11].

In this paper we present our experience with building a prototype of SMMART – **S**ystem for **M**obile **M**arketing: **A**daptive, **p**eRsonalized and **T**argeted. SMMART is a context-aware application, delivering narrowly targeted promotions to the users wireless mobile devices, such as PDAs and smart phones, when they are in a close proximity or inside a retail store. SMMART adapts to the needs of its user by unobtru-

sively monitoring his/her shopping habits and learning the user's personal preferences. The functionality of the system may be described using a simple metaphor:

*Whenever you go to a retail store, there are brochures advertising current sales. You may be interested in some products, but have no time or intention to look through all pages in the brochure with no guarantee of finding anything interesting. Imagine that there is a genie who knows all about your shopping interests that will carefully read the entire brochure and clip only those promotions that precisely match your interests. SMMART is that genie running on your PDA or smart phone, which can work at any retail store equipped with the corresponding technology. Moreover, it will know when your interests change as long as you continue using it.*

This paper is organized as follows: Section 2 describes related work; Section 3 described a scenario of using SMMART; Section 4 discusses SMMART architecture; Section 5 concludes the paper and presents possible directions of future work.

## 2 Background and Related Work

Varshney and Vetter [12] provide a classification of m-commerce applications, which includes a category of mobile advertising applications that typically use demographic or other information specified by the consumers to deliver targeted advertising messages [14]. These applications may be location-sensitive, delivering the message only to the users that are located in the vicinity of the retailer being advertised [11]. However, coverage area of such applications depends on the precision of the user location determined by the network technology used for wireless connectivity. Each mobile advertising application should cover a small area and narrowly target its recipients to avoid network congestion and overwhelming consumers with a large number of irrelevant advertising messages.

A context-aware system operates and adapts itself based on the knowledge about its user's state and physical surroundings [12]. One of the methods to obtain location context without gathering precise geospatial data is by detecting a connection to a wireless personal area network (PAN), such as WiFi or Bluetooth. Context-aware services enabled by PAN technologies can only reach customers located within a close physical proximity of the wireless service provider. eNcentive framework described in [10] is a context-aware m-commerce application used to distribute electronic coupons. However, it pushes all available coupons to its users regardless of their preferences. To be effective, eNcentive is deployed at a large number of retail sites and requires an even larger number of customers carrying wireless PDAs.

SMMART belongs to the same class of applications as eNcentive and is used to deliver targeted marketing information to customers whose preferences match products that are currently on sale at retail stores. SMMART guarantees a high level of privacy because it does not transmit any personally identifiable information and cannot be used by retailers to track their customers and their buying habits.

SMMART is an example of a user-centric, context-aware pervasive system. In general, pervasive computing systems have the following characteristics [5]: ubiquitous access, context awareness, intelligent behavior, and natural interaction. Ubiquitous access and ubiquitous computing, introduced by Weiser [13], refer to an environment where users are surrounded by computational power and applications. Sen-

sors, smart phones, pagers, PDAs, different miniaturized and embedded devices are the hardware environment supporting ubiquitous services and applications.

Ubiquitous services are context-aware in the terms of location-awareness, time awareness, device-awareness and personalization [6]. Context-awareness refers to the ability of a system to recognize users, to interpret context information and to run in an appropriate fashion for the users, applications and services.

Intelligence in pervasive computing comprises adapting to user behavior, personalization of application and services and supplying users with information at the right place and time. SMMART is designed and implemented as an intelligent pervasive system with abilities to adapt, to target user with information and to personalize its services. SMMART incorporates a number of features enabling it to adapt its behavior based on the current context and past user input.

Natural interaction in pervasive computing refers to system modality where the same functionality is delivered through voice (speech recognition and synthesis), wireless interfaces, and gesture recognition. SMMART is designed to be extensible with a provision to eventually develop a multimodal voice-enabled interface with speech recognition and synthesis.

### 3 Using SMMART

In this section we present a scenario of how a hypothetical shopper named Bob could use SMMART in his everyday shopping (Fig. 1).

Bob recently installed SMMART Client software onto his wireless PDA, entered his musical preferences, as shown in Fig. 2a, and drove to his favorite place to buy CDs. As Bob enters the store, his SMMART Client makes a connection with the store's SMMART Server and tells the server about his preferences. The server responds with a list of products that match Bob's preferences and are currently on sale, as shown in Fig. 2b. Bob selects *Every Breath You Take* on the screen of his PDA to view more information about the promotion. As Bob clicks on this product, his SMMART Client assumes that he may be interested in other products by *Police* and its musicians. In this case, keyword *Sting* (the lead singer of *Police*) is automatically added into the list of Bob's preferences. Bob also decides to purchase *On Every Street* by *Dire Straits*. Clicking on this product description has two consequences: Bob's interest in *Dire Straits* is confirmed and the keyword *Mark Knopfler* (the founder of *Dire Straits*) is added to his preferences.

Later Bob decides to visit a bookstore. At this moment, Bob's preferences include five keywords, as shown in the lower portion of Fig. 1. Upon entering the store, his

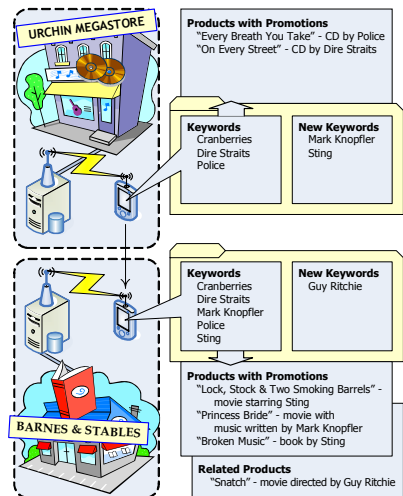


Fig. 1. A typical scenario of using SMMART

SMMART Client receives all current promotions matching his interests, presenting Bob with three products. Bob is most interested in *Lock, Stock and Two Smoking Barrels*, a movie starring *Sting*. As Bob selects this product description, Bob’s interest in *Sting* is confirmed and his preferences are updated with a new keyword – *Guy Ritchie* (director of this movie). Bob’s SMMART Client also offers a list of related products, which include *Snatch*, a movie directed by *Guy Ritchie*.

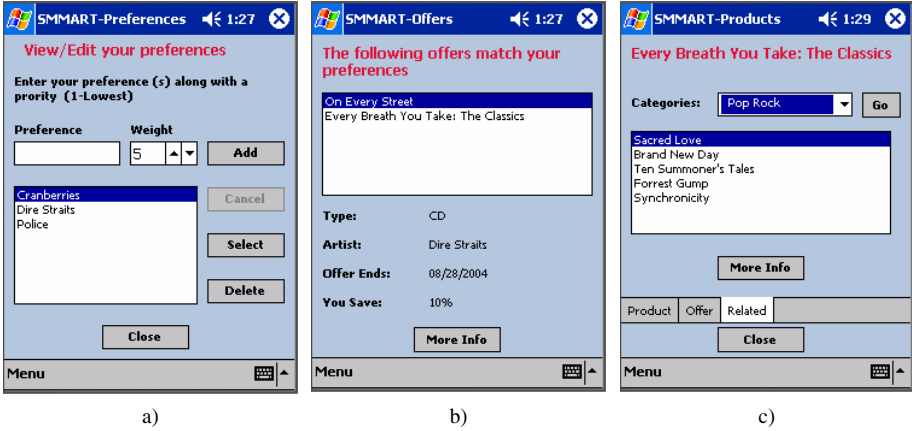


Fig. 2

## 4 Overview of SMMART Architecture

As illustrated by the above scenario, SMMART consists of a server installed at every participating retail location and clients for mobile wireless devices that pull information from the server (Fig. 3). An inventory database of a retail store provides the basis for all data available to the SMMART Server. Product Manager retrieves all relevant information about a specific product, which is then used by SMMART Client when the user chooses to view the details about a particular offering. Similarly, Promotion Manager retrieves all promotion information for a given product. Search & Match Agent is the core of the SMMART Server. This agent receives a list of keywords from the client ordered by their relevancy to the user’s interests. For each keyword in the list, the agent finds all matching products that currently have a promotion and adds them to the result. The result consisting of matching products is sorted in the order of relevance to the user preferences and returned to the client.

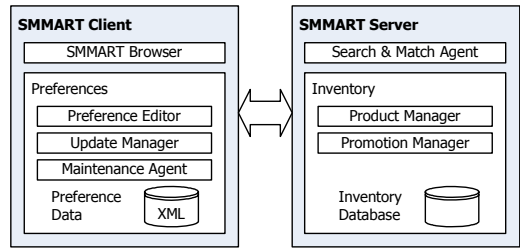


Fig. 3. Architecture of SMMART



As shown in Fig. 3, SMMART Client consists of two main components: SMMART Browser and Preferences module. Typically, after SMMART Client connects to and communicates with a server, the user is presented with a list of matching products that currently have promotions (Fig. 2b). The user explores each product in detail by viewing its full description in SMMART Browser. This information is divided into several sections: product information, offer details and related products (Fig. 2c), which can also be explored using the SMMART Browser.

Preferences module contains and manipulates Preference Data – a list of keywords stored in XML format. Each keyword is associated with a numeric weight representing its relevancy to the user’s interests and a timestamp indicating the last time this keyword or its weight was updated. As described below, the keyword weights and timestamps also facilitate the learning capabilities of SMMART.

When the user runs SMMART Client, the Preference Data is scanned. If no keywords are found, the user is prompted to add new keywords using Preference Editor (Fig. 2a). Main responsibilities of this module include adding new, editing and deleting existing keywords, manually changing the weights of existing keywords if needed.

When the user indicates an interest in a particular product by viewing its full description in SMMART Browser, Preference Data is automatically updated by Update Manager. If the keywords associated with a product are already present in Preference Data, their weights are incremented. Otherwise, they are added with a default weight.

SMMART assumes that by viewing full product description the user signifies his interest the product. It is possible for keywords not reflecting true user interests to be automatically added to Preference Data. The primary goal of Maintenance Agent is to detect and remove any keywords that are erroneous or represent past interests of the user. Maintenance Agent runs at application startup and looks for keywords that have not been updated within a specified period and decreases their weight. Eventually, such keywords will be placed in the Recycle Bin and will not be used to find matching products.

## 5 Implementation of SMMART Prototype: Lessons Learned

To prove the functional feasibility of the SMMART, we implemented its fully operational prototype. We chose C# and ASP.NET to implement SMMART Server running on Microsoft .NET Framework. SMMART Client is also implemented in C# running on Microsoft .NET Compact Framework. Our inventory database runs on SQL Server 2000. All tests were conducted using Dell Axim PDAs. The remainder of this section discusses different approaches to the specific implementation details of a context-aware mobile marketing application, such as SMMART, as well as some challenges that we faced in that process.

### 5.1 SMMART Context Information

SMMART uses several types of context information [2] as described below.

**Physical location context** is needed to determine which store’s inventory is to be searched every time a user wishes to use SMMART. Our application does not require the knowledge of geographical coordinates of the user’s location. Instead, we are using the information about the physical proximity of the user (SMMART client) to

the store (corresponding server). Our approach to obtaining this type of context information is discussed in the next section.

**User context** (user identity) determines what specific information is presented to the shopper. For example, it is reasonable to expect that two different users of SMMART visiting the same store will see a different set of offers because they have different their shopping preferences. In SMMART, user context is also affected by the previous experiences of each user. In the beginning of using SMMART, it is the user's responsibility to enter some keywords describing his or her shopping preferences (if no preferences are entered, the user will have an option to see all offers available at a store). As soon as the user begins browsing through the available offers, SMMART starts analyzing the user's browsing patterns by matching and updating keywords describing each viewed product and keywords in the Preference Data.

## 5.2 Infrastructure of Client-Server Communication

XML Web Services appear to be a good choice for the logical structure of SMMART client-server communication [9]. Firstly, web services fit well in the general philosophy of SMMART: a number of functionally and semantically related methods are united under the umbrella of a single service; all methods work with the same data, i.e. the store's inventory. Secondly, using web services helps overcome the burden of possible network disconnections due to the statelessness of the connection. However, XML and SOAP add a sizeable overhead to the amount of the exchanged data [1], which may result in a delayed application response and congestion of the wireless network connection.

Our primary objective was to prove the viability of SMMART concept. Current implementation of the prototype uses XML web services over a Wi-Fi wireless LAN. Such a choice of technologies works well for a large class of existing Wi-Fi-enabled PDAs. This also assumes that each SMMART site must be equipped with one or more wireless access points. Also, each site must route all network traffic from an access point only to the web server hosting SMMART web services. This enables an unambiguous identification of the store to which a SMMART Client is connected. Additionally, such a routing scheme prevents possible hijacking of the wireless bandwidth. However, there is another class of devices, which includes smart phones equipped with Bluetooth sensors requiring a different combination of network and data access technologies. SMMART can easily be implemented to work with this set of client hardware with no modifications to the architecture of the system.

## 5.3 Design for Handheld Devices

Designing applications for handheld devices is greatly influenced by their hardware limitations, primarily small screen, slow CPU, small amount of RAM, and short battery life. In an application such as SMMART, only the most essential information must be displayed on the screen. This is not only because the screen is small, but also because this application is used in an environment where the user may be easily distracted by many environmental factors. SMMART requires minimal data processing since its algorithms are simple and produce no noticeable delay on Dell Axim PDAs, on which SMMART prototype was tested.

The most challenging issue in designing a networked application for a PDA is the short battery life. Currently, maintaining a WiFi connection on a PDA is a very energy-consuming task. A SMMART client requires wireless connectivity for browsing, searching or matching of any products in the store inventory. A connection is not required for editing of preferences. However, while running a SMMART client, a PDA can be powered off at any moment. When it is turned back on, possibly at a different location, a running SMMART client will detect the changes in the wireless network, find an available SMMART server and obtain a new set of products matching the user's preferences.

#### **5.4 End-User Acceptance**

When a new application arrives on the market, it is crucial to know whether end-users will find it intuitive and easy to use. If the users do not want to use the system, it does not matter how technologically advanced it is or how much savings it could yield. We conducted a survey of potential end-users of SMMART who were given an opportunity to test its prototype using their own shopping preferences in our "test store" containing about one hundred products. The results of our survey indicate that its participants have a very favorable opinion about SMMART. Specifically, based on their own experience with SMMART, 80% of the survey respondents agreed that the system makes good matches between their shopping preferences and products in the test store. Given a chance to browse through the products found as a result of matching of preferences or searching for keywords, navigate through the different screens of the user interface and system options, 80% of the respondents agreed that interface of SMMART Client is intuitive and easy to use. Finally, 93% of the respondents said that if they owned a mobile device running a SMMART Client, they would be willing to use the system in their everyday shopping.

Consumers are always concerned about their privacy: why would they give away potentially compromising information about themselves and their preferences? In terms of preserving the user's privacy, using SMMART is equivalent to searching the inventory of a store with an Internet portal. In this process, the store can deduce the consumer's interest in certain products. A typical online store can also easily detect whether a particular search resulted in a purchase. SMMART enables consumers to make such searches completely anonymous because stores cannot make a connection between a search and a purchase. Additionally, while performing a search, SMMART filters and sorts the obtained results according to the criteria of their relevance to multiple keywords. This effectively eliminates the necessity to reformulate the search query, which arises frequently in searching the inventories of online stores.

#### **5.5 Retailer Acceptance**

Increased revenue is the primary factor that determines the acceptance of SMMART by retailers. Deploying SMMART at a single retail store or at a chain of affiliated stores must be economically justified. The costs of the framework, its supporting infrastructure, data upkeep and maintenance must be less than the revenue from additional sales generated by the customers using SMMART.

At the same time, retailers should not view SMMART as a potential tool to drive up the competition. It is in the retailers' best interests not to allow shoppers to com-

pare products easily, but rather to distinguish their products from the competitors, which can be achieved through personalization. SMMART is designed for use at only one store at a time and therefore shoppers will be unable to compare prices among different stores. This feature should be appealing to the retailers because it creates an easy way to automatically create personalized shopping lists without any investments in additional demographic and market research.

To demonstrate the increase in revenues, we created a simulation model, in which we measured a relative increase in sales generated by purchases resulted from product matches and recommendations made by SMMART. Our experimental results show that SMMART yields the highest increase in sales with the low values of  $P(c)$ , the metric we used in our model, which represents the probability that a customer  $c$  would make a purchase uninfluenced by SMMART. This is typical for upscale stores in shopping malls, stores that sell large ticket items, or stores where people come to socialize, as well as to shop. For example, according to our data, when the probability of a customer to make a purchase is 20% and when only 5% of all customers are carrying SMMART-enabled mobile devices ( $S(c) = 5\%$ ), using the system would yield an almost 13% increase in sales. Alternatively, with higher values of  $P(c)$ , which are typical for stores where customers are determined to make a purchase and stores where customers make routine purchases, such as grocery stores and supermarkets, the expected impact of SMMART is more modest. With  $P(c) = 90\%$  and  $S(c) = 5\%$ , SMMART yields slightly less than a 3% increase in sales.

## 5.6 The Big Picture

Following our experience with SMMART, we propose a generic client-server architecture for context-aware systems that subsumes a number of other architectures proposed in the literature [3, 8]. Our architecture comprises four core components: sensor information and drivers, context client, context server, and context database.

Different sensors remotely or locally connected to the server, usually network-based, are responsible to supply context information – location, time, device or object status, and personalization. The drivers are software components that interpret sensor information and convert it in appropriate context information for Context Interpreter. Some authors called the drivers widgets [4], or adaptors [6].

Context Interpreter is responsible for converting context information received from the drivers or from the context client input to higher levels of context information understandable by an application and its services. For example, physical coordinates can be converted to street name and number and/or building and floor. If the context can be recognized and interpreted, the context is transferred to the Application Manager, which runs an application or applications connected to the current context. If the context cannot be recognized because of ambiguous, insufficient or inaccurate information, the Context Interpreter queries the context tables and uses context rules in attempt to find the right context. The context tables contain user context information: user preferences, user habits, past user schedules and activities. The context rules are similar to knowledge database rules. Context Interpreter acts as an inference engine to find the right context and to deliver it to the Application Manager.

Application Manager matches context information to applications and services and initiates their execution controlling the running and stopping of services and

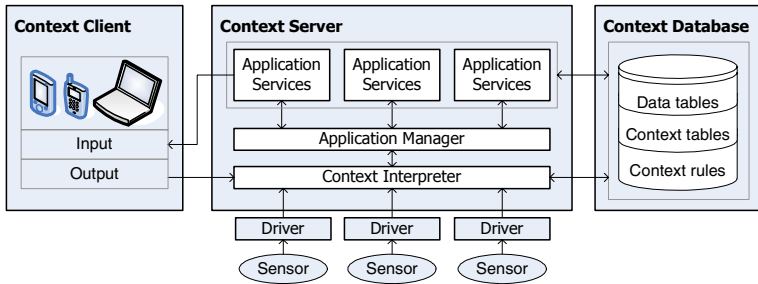


Fig. 4. Generic Architecture for Context-aware Systems

applications. The applications can query the data tables of the context database to retrieve information and to deliver it to the clients as output.

The services are components that execute actions on behalf of applications (for example turning the lights on or delivering notifications). Usually they are implemented as software agents. The services proactively monitor user calendars and schedules, email accounts, and deliver information to the user in a timely manner.

## 6 Conclusion and Future Work

In this paper we presented a novel approach to mobile marketing in context-aware environments. We built a prototype of SMMART implementing this approach. A user of a wireless PDA equipped with SMMART can receive promotions from retail stores for those products that match his interests. We also studied the economic feasibility of SMMART that indicate that it will be extremely effective in stores where customers need additional incentives to make purchases. Possible examples include stores in shopping malls, bookstores, consumer electronics warehouses, and any other retailers where consumers come not only to shop, but also to socialize.

SMMART can be extended by providing more features that would enhance its usability. A product inventory search would allow a user to search the entire inventory of a store. Product information pages of SMMART Browser could be enhanced with a map schematically showing the location of the selected product in the store. SMMART could also provide the user with the ability to reserve an item which is on back order at the sale price or the option of ordering an in stock item to be picked up and purchased on a designated date and time. This feature could also work well with large items such as big screen TV's or other products where the inventory is not kept on the display floor. Finally, SMMART could be extended with a multimodal interface giving the users an ability of voice communication with the system.

## References

1. H. Chu, C. You, C. Teng. "Challenges: Wireless Web Services," In Proceedings of 10<sup>th</sup> International Conference on Parallel and Distributed Systems (ICPADS'04), July 7-9, 2004, Newport Beach, CA.
2. Dey. "Understanding and Using Context" in Personal and Ubiquitous Computing, Vol. 5, No. 1, pp. 4-7, Springer-Verlag 2001.

3. Dey, G. Abowd. "The Context Toolkit: Aiding the Development of Context-Aware Applications," In Proceedings of the Workshop on Software Engineering for Wearable and Pervasive Computing, Limerick, Ireland, June 6, 2000.
4. Dey, G. Abowd, and D. Sabler. "A Context-Based Infrastructure for Smart Environments," In Proceedings of the 1<sup>st</sup> International Workshop on Managing Interactions in Smart Environments, Dublin, Ireland, Dec. 13-14, 1999.
5. Fersha. "Coordination in Pervasive Computing Environments," In Proceedings of 12<sup>th</sup> IEEE International Workshop on Enabling Technologies, 2003.
6. T. Hofer et al, "Context-Awareness on Mobile Devices – the Hydrogen Approach," Proceedings of 36<sup>th</sup> Hawaii International Conference on System Sciences, 6-9 January, 2003.
7. P. Kannan, A. Chang, A. Whinston. "Wireless Commerce: Marketing Issues and Possibilities," In Proceedings of The 34<sup>th</sup> Hawaii International Conference on System Sciences, IEEE CS Press, 2001.
8. S. Meyer, A. Rakotonirainy. "A Survey of Research on Context-Aware Homes," In Proceedings of the Workshop on Wearable, Invisible, Context-Aware, Pervasive and Ubiquitous Computing, Adelaide, Australia, 2003.
9. T. Pilioura, T. Tsalgatidou, S. Hadjiefthymiades. "Scenarios of Using Web Services in M-Commerce." ACM SIGecom Exchanges. Vol. 3, No. 4, Jan. 2003, pp. 28-36.
10. O. Ratsimor, T. Finin, A. Joshi, Y. Yesha. "eNcentive: A Framework for Intelligent Marketing in Mobile Peer-to-Peer Environments," In Proceedings of The 5<sup>th</sup> International Conference on Electronic Commerce (ICEC-03), Pittsburg, PA, October 2003.
11. U. Varshney. "Location Management for Mobile Commerce: Applications in Wireless Internet Environment." ACM Transactions on Internet Technology. Vol. 3, No. 3, Aug. 2003, pp. 236-255.
12. U. Varshney, R. Vetter. "Mobile Commerce: Framework, Applications and Networking Support." Mobile Networks and Applications, Vol 7, pp. 185-198, Kluwer, 2002.
13. M.Weiser. "The Computers for the 21st Century", Scientific American, 265, 3, September 1991.
14. H. Yunos, J. Gao, S. Shim. "Wireless Advertising's Challenges and Opportunities." IEEE Computer, Vol. 36, No. 5, pp. 30-37, IEEE CS Press, 2003.

# Ubiquitous Organizational Information Service Framework for Large Scale Intelligent Environments

Kwang-il Hwang<sup>1</sup>, Won-hee Lee<sup>1</sup>, Seok-hwan Kim<sup>1</sup>,  
Doo-seop Eom<sup>1</sup>, and Kyeong Hur<sup>2</sup>

<sup>1</sup> Department of Electronics and Computer Engineering, Korea University,  
#1-5ga, Anam-dong, Sungbuk-gu, Seoul, Korea  
{brightday, wangpepe, sukka, eomds}@final.korea.ac.kr

<sup>2</sup> Department of Computer Education,  
Gyeongin National University of Education  
khur@ginue.ac.kr

**Abstract.** In this paper, we introduce a concrete, practical Ubiquitous Organizational Information (UOI) service framework, providing novice users intelligent and useful services with respect to the environment. The UOI framework based on the sensor networks is composed of 3-level hierarchical network architecture. To provide a rich array of services, the modular software framework and foundation software are designed and implemented on our hardware prototype. We define three representative UOI services and illustrate each service flow operating on the proposed UOI network. In addition, we describe some details in the implementation of a distributed UOI network on the UOI test-bed.

**Keywords:** Distributed Sensor Networks, Embedded Systems, Intelligent Environment, and Ubiquitous Computing.

## 1 Introduction

To coincide with the grand pervasive computing vision, everyday computing spaces will need to become a component of the user's normal background environment, gradually becoming more ubiquitous in nature. Mark Weiser first initiated the notion of Ubiquitous Computing at Xerox PARC [1], who envisioned in the upcoming future, ubiquitous interconnected computing devices that could be accessed from any location, used effortlessly, and operate unobtrusively, even without people's notice of them, just as that of electricity or telephones are used today.

Many researchers define an *intelligent environment*, as one of the most representative applications of Ubicomp, as an augmented spacious environment populated with many sensors, actuators and computing devices. These components are interwoven and integrated into a distributed computing system, capable of

perceiving context through sensors, to execute intelligent logic on computing devices and serve occupants by actuators. This intelligent environment is extending its range from a users' personal room or classroom, to a large house or building.

Let us suppose the following situation. We visit an unfamiliar environment, which presents a wide area, such as an amusement park, university campus, or large building. Confusions often arise when finding the location of something or where to travel next. Furthermore, the use of certain facilities may be desired or someone in the organization may need to be found. Such users' needs will be satisfied with an intelligent service involving information regarding the organization. Our UOI service framework is designed to provide such an intelligent service to users, especially for large scale environment. In this paper, the presented UOI service framework presents more concrete and practical way to create intelligent environment.

The rest of this paper is organized as follows. We first outline several researches related to the intelligent environment. Then, the UOI framework is presented, which is composed of three major components. Subsequently, the UOI service flow through the distributed UOI network is illustrated. Our hardware prototype and UOI foundation software operated on the prototype is also introduced. Lastly, details in the implementation of a distributed UOI network are described. A conclusion is provided with a description of future work.

## 2 Related Work

There have been substantial researches relating to the construction of ubiquitous environments.

Cooltown [2] and the associated CoolBase infrastructure aim to give people, places, and things a Web presence. Although Web technology is proven and widely available, it has inherent complexity, since, to be connected to the Web, a fully supported TCP/IP stack and system capable of running the relatively heavy software is required.

Projects, such as Gaia [8], Microsoft Easy Living [3], and CORTEX [9], aim to develop an infrastructure to support augmented environments in a fairly broad sense. They provide basic abstractions and mechanisms for coping with the dynamics and device heterogeneity of pervasive computing environments. There is quite a large difference between the projects and the framework presented in this paper. While they provide application models that are still rather generic, our work supports a rather specific application model.

In such a sense, PiNet [10] is the most similar to the presented model in that the final goal is to provide an organizational information service to users. However, the work in this paper is distinguished from PiNet primarily in the uses of sensor networks. In contrast to PiNet using a global cellular network as an infrastructure, the UOI adopts distributed sensor networks. In our research, service network infrastructure and service framework based on sensor networks, are more emphasized, instead of focusing on user perception or virtual reality as in [4 - 7].



### 3 Architecture of UOI Framework

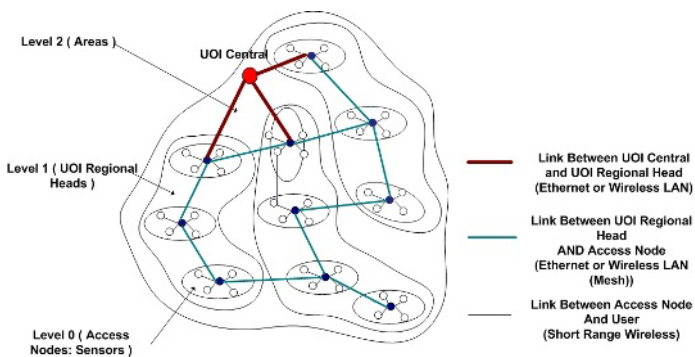
The UOI service framework infrastructure is based on distributed sensor networks. It is assumed that the environment is covered with innumerable tiny sensor nodes, which are extremely limited in power, processing, and memory resources. The sensor nodes are called access nodes, since they are used as access points connecting users to a UOI network. Each node is aware of its own location information by manual planning or other localization algorithms, and possesses the ability to communicate with user's devices via short range wireless communication. The UOI service infrastructure using sensor networks, not global networks such as cellular networks or GPS, presents advantages as follows.

- Guarantee of freshness with respect to the dynamics of information
- Security
- Service charge issues regarding information use

Firstly, the update of information with respect to the change of an organizational structure or service will be achieved faster and easier through a scalable UOI framework. Secondly, organizational information will be safer from outside networks. Lastly, users are allowed to use the service without any extra communication charge. In addition to these advantages, compared to WLAN networks, the UOI network architecture provides more elaborate location information and increases efficiency in the use of facilities through sensors and actuators. The UOI network architecture also enables localized information processing and fusion, by clustering regionally adjacent access nodes.

#### 3.1 Key Features of UOI Framework

The presented framework is designed to efficiently provide UOI services. Such UOI framework includes the following array of features.



**Fig. 1.** Hierarchical UOI network architecture based on distributed sensor networks

**Hierarchical architecture:** As shown in Fig. 1, the UOI framework is composed of three-tier architecture, more specifically, access node level, regional head level, and UOI central. Regionally adjacent nodes form a cluster in which a cluster head node is responsible for managing its own cluster member nodes. In addition, the clusters form a network of tree-based or mesh-based topology to communicate with each other. The clusters are also connected to the UOI central, which operates similarly to a central server. This hierarchical architecture of UOI framework makes it possible to localize information in a cluster, and reduces traffic by using aggregation and fusion within a cluster head. This feature is also useful when the environment is managed regionally.

**Property-based naming and information-centric routing:** The UOI platform uses property-based naming, similar in concept to naming in the Directed Diffusion [11], not global ID such as IP address or MAC address, as a node identifier. Each node has an inherent name related to its property such as location or sensing ability. For example, ‘East2 Floor1’ means the node is the 2<sup>nd</sup> node from UOI regional head of the 1<sup>st</sup> floor to east. In addition, ‘Tb21 KoreanRest1’ will be the 21th table number of a Korean restaurant in a huge amusement park. Information-centric routing is also enabled by virtue of the property-based naming, which is different from address-centric routing.

**Distributed querying and tasking:** In the UOI framework, user’s service request is translated into a query to be flooded to the UOI network. The query is injected in each access node and the UOI regional head via the UOI hierarchy. In each node, the query generates a corresponding task, operating on the UOI foundation software. This distributed tasking demonstrates some results with respect to the query and only the nodes, which have data matched with the query, can report matched information to UOI central. This feature reinforces the distributed information processing ability in the UOI network, in contrast to other global or centralized networks.

**Transparent services:** The UOI framework provides transparent services to users. Users only request a service with their device, and specific actions for configuration or registration are not required. When entering an area covered with a UOI network, the user is expected to turn on the device and be automatically connected. No configuration changes are necessary as the user moves from one site to another. The network needs no pre-knowledge regarding the device attempting to connect to it.

### 3.2 Components for UOI Service Framework

The UOI network infrastructure is composed of three distinguished components, *Access node*, *UOI regional head*, and *UOI central*, as shown in Fig. 2. These independent components play an important role in building a UOI service framework with a hierarchical architecture.

*Access node* is the most basic component, allowing users to access the UOI infrastructure. This component is composed of UOI foundation software, Query Translation Unit, Task Manager Unit, User Interaction Manager, and Location Management Unit.

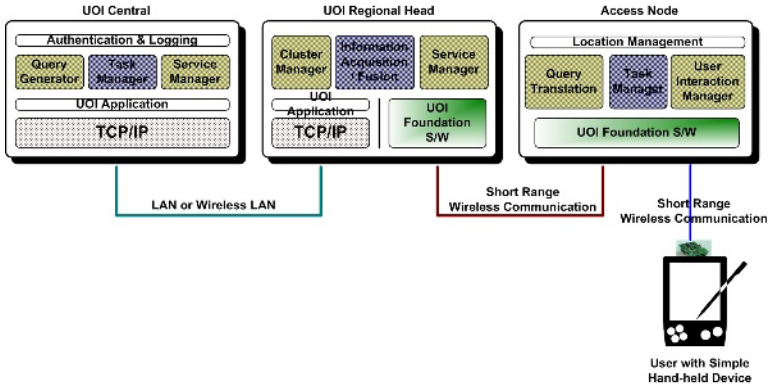


Fig. 2. Components for UOI service framework

*UOI Regional Head* as a cluster head is responsible for managing access nodes in its own cluster, and operates as a gateway between access nodes and UOI central. The UOI head includes two independent networks software: TCP/IP for connecting to UOI central by LAN or WLAN, and UOI foundation software for managing cluster members. In addition, the role of UOI head is performed with UOI application, Cluster Manager Unit, Information Acquisition and Fusion Unit, and Service Manger Unit.

*UOI Central* represents component of the highest level in the UOI service hierarchy. It plays an important role in generating queries with respect to user's service request, managing tasks and services and authenticating users and logging. The works are performed with the Query Generation Unit, Task Manager, Service Manager, and Authentication Unit.

### 3.3 Organizational Information Services Through Distributed UOI Networks

Users entering an unfamiliar environment want to get some organizational information and be available to freely use various facilities without pre-knowledge in the organization. Furthermore, users do not want to manually configure or register themselves to use organization services. We summarize the UOI services fulfilling such user's requirements into the following service category.

**Location guide service:** First of all, the most basic service offered to novice users is location guidance. Generally, guiding services using GPS are the most common. However, in GPS, the service with high resolution is not guaranteed. In addition, GPS is difficult to be used for indoor location systems, such as large buildings.

Compared with the GPS service, our UOI framework provides more reliable location guide service with higher resolution through distributed UOI networks. Figure 3 shows the procedure of location guide services in the UOI framework. As shown in Fig. 3, the nearest access node listens to user's service request and then

sends the request message to the UOI regional head. After successful authentication between the head and central, the user’s service request at UOI central is transformed into a corresponding query and the query is flooded over the network. As soon as the query is received, each node executes a corresponding task from the query. The nodes with matched data send the response upstream, immediately. UOI central performs aggregation, fusion, and result generation during a given period. The result is delivered to the UOI regional head of the targeted node.

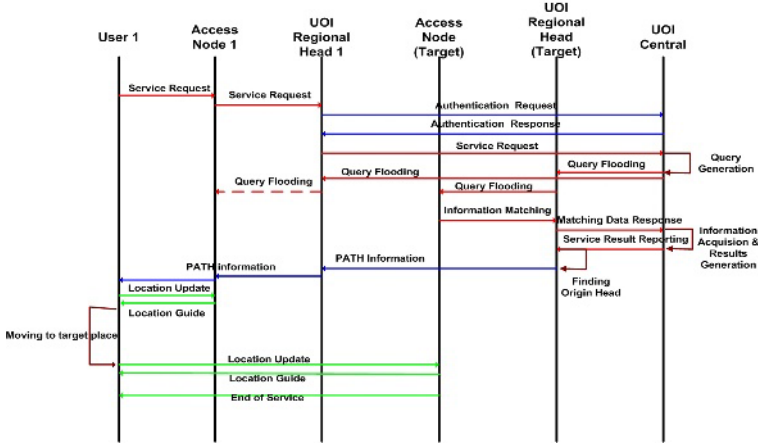


Fig. 3. Process procedure of location guide service in UOI framework

The targeted head retrieves the source head to establish an exclusive link between the two regional heads. The targeted head sends PATH Information to the source head, including all path information from the target to source. Note that this logical path is regarded as a reliable physical path, since we assume the environment is densely covered with sensor nodes. Simultaneously, all access nodes, included on the path between two heads, execute the location guide task and process location update messages from the user. Now, a user can view the location where the user wants to go, which will be displayed on his or her device. The user’s device sends location update message periodically and each access node guide the user by responding with a location guided message.

**Use of facilities in the organization:** One of the representative intelligent services is to allow users to use various facilities without prior knowledge in the organization. For example, in an amusement park, we want to find a specific amusement facility. However, generally we must wait for a long time just for the short instance actually using the service. This delay may annoy users, but they cannot help using the service.

However, in the UOI service framework, users do not have to wait for a long line. Instead of waiting until the user’s order comes, users simply make a reservation for

the use of the facility that the user wants while enjoying other amusement facility. After the reservation, when the user's service is available, the user is alarmed through the UOI network. Figure 4 shows the detailed flow for user to use facilities in the UOI framework. The service is divided into four phases. The first phase consists of service request and authentication, and second phase includes the discovery of service access points and a status result report. These two phases are almost similar in a location guide service. The third phase is user's reservation to use the service. The last is service reservation delivery and result reporting. All the processes are accomplished through distributed UOI networks and completely transparent to users.

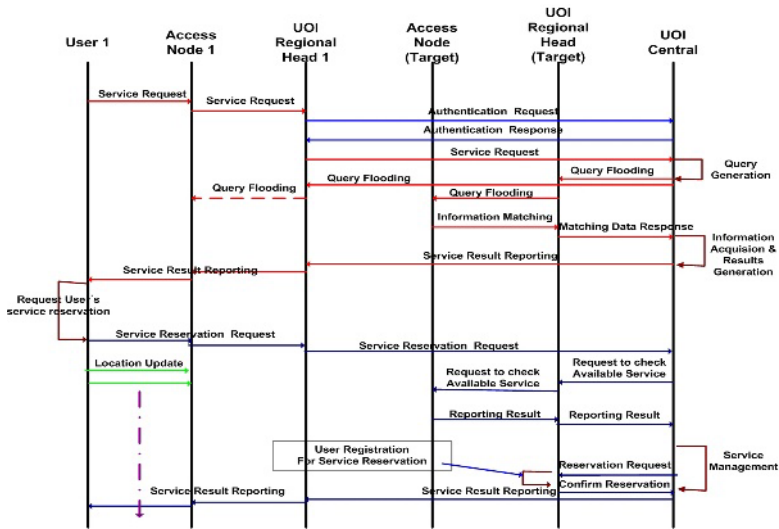


Fig. 4. Process procedure for use of facilities in UOI framework

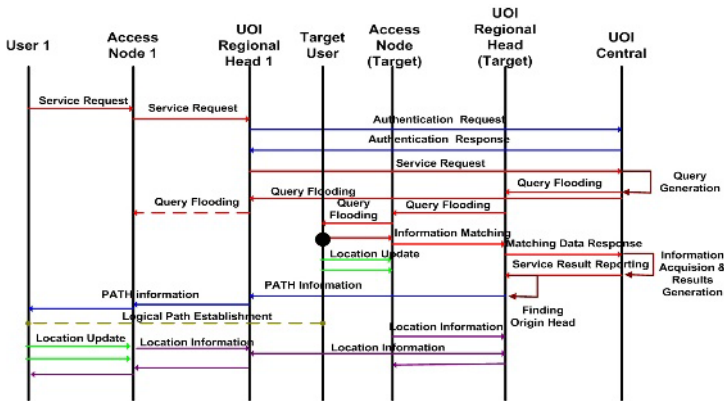


Fig. 5. Process procedure for people search

**People search:** The other required service involves finding people within an organization. This is useful to find out a missing child’s location or the location of someone who is not available to use a phone in the organization. In addition, this service is used to retrieve the corresponding user, who is reserved to use a facility, to notify that the service is now available. For this service, general steps are similar to others but it is outstanding that the exclusive logical path between the user, who requests the service, and target user is established. Note that for concurrent mobility support, location managements for both are performed as shown in Fig. 5.

### 4 UOI Network Implementation

In order to implement the proposed UOI service framework, a UOI test-bed as shown in Fig. 8, was developed. As an access node, a scalable prototype having a very compact size of a square inch was developed as shown in Fig. 6. Also, Figure 7 illustrates the event-driven UOI foundation software operated on the prototype.

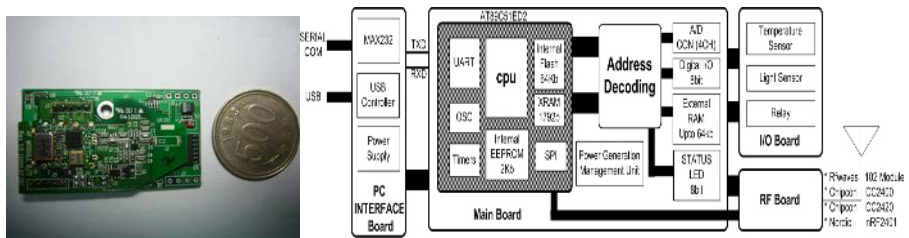


Fig. 6. Prototype for UOI access nodes and user interface

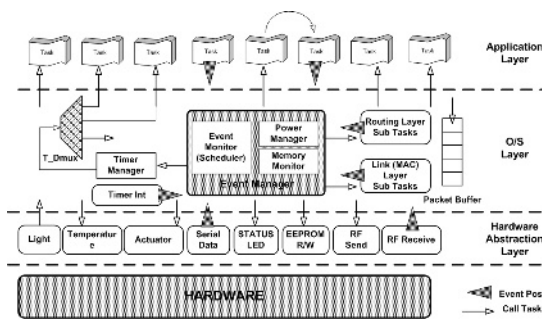


Fig. 7. UOI foundation software architecture

The Test-bed is composed of access nodes, which is built-in UOI foundation and framework software on our prototype, regional heads, and UOI central. For higher performance of Regional heads, an embedded system with Intel’s Xscale MCU

(PXA255) was used as a UOI regional head and UOI central software was executed on a desktop PC. The deployment of each component is as shown in Fig. 8, including two users and two service points.

In the test-bed, three kinds of UOI services that we described in Section 3.3 were experimented: in the first experiment, user A is guided to service point 2 through the UOI service. Secondly, user A and B reserved an available arbitrary seat in service point (room) 1 and 2, respectively, and then were guided to the corresponding service point. In the final experiment, mobile user A tracked mobile user B, continuously.

Initial architecture design was revised from a more practical sense through the experiment. After experiencing innumerable trials and errors with the experiment, measurable improvements were made to the complete UOI service framework architecture.

## 5 Conclusion and Future Works

We proposed a concrete, practical Ubiquitous Organizational Information (UOI) service framework, providing novice users intelligent and useful services respecting the environment. The UOI framework consists of hierarchical network architecture, based on distributed sensor networks. To provide a rich array of services, the designed UOI framework and foundation software are implemented on our hardware prototype. In addition, representative UOI services were tested on the UOI test-bed.

Currently, we are investigating to extend the kind of services and improve the quality of services.

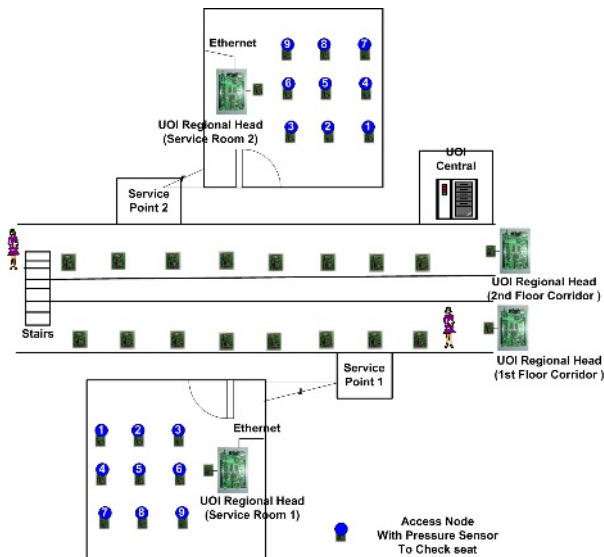


Fig. 8. Test-bed environment for implementation of UOI service framework

## References

1. M. Weiser, "The Computer for the 21<sup>st</sup> Century," *Scientific American*, September, 1991.
2. HP Cooltown. <http://www.cooltown.com/cooltown/>
3. Microsoft Easy Living. <http://research.microsoft.com/easyliving/>
4. MIT AI Lab AIRI Project, 2000. <http://aire.csail.mit.edu/>
5. Georgia Tech, Aware Home Research Initiative (AHRI) Project, <http://www.cc.gatech.edu/fce/ahri/>
6. CMU Project AURA. <http://www-2.cs.cmu.edu/~aura/>
7. Sony Augmented Surfaces. <http://www.csl.sony.co.jp/person/rekimoto/as/>
8. M. Roman, C. Hess and R. Campbell, Gaia: An OO middleware infrastructure for ubiquitous computing environments, *ECOPOOOSWS 2002*, 2002.
9. P. Verissimo, V. Cahill, A. Casimiro, K. Cheverst, A. Friday and J. Kaiser, CORTEX: Towards supporting autonomous and cooperating sentient entities, *European Wireless 2002*, Florence, Italy (February 2002).
10. Carmel, B. et al., "PiNet: Wireless Connectivity for Organizational Information Access Using Light-weight Handheld Devices," *IEEE Personal Communications*, Aug. 2001.
11. C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," *6th ACM/IEEE Mobicom*, 2000.



# TS-U: Temporal-Spatial Methodology for Application Checking of the Systems in the Ubiquitous Environment

Fran Jarnjak<sup>1</sup>, Jinhyung Kim<sup>1</sup>, Yixin Jing<sup>1</sup>, Hoh Peter In<sup>1</sup>,  
Dongwon Jeong<sup>2</sup>, and Doo-Kwon Baik<sup>1</sup>

<sup>1</sup> Dept. of Computer Science and Engineering, Korea University, Seoul, Korea  
{fran, koolmania, jing, hoh\_in, baikdk}@korea.ac.kr

<sup>2</sup> Dept. of Informatics & Statistics, Kunsan National University, Gunsan, Korea  
djeong@kunsan.ac.kr

**Abstract.** In the paper a novel methodology, TS-U, is proposed based on existing model checking techniques that were extended to successfully model the systems in a ubiquitous environment by introducing time and space constraints relevant in modeling of some ubiquitous system for its validation. Method proposed requires only slight modification of the existing model checking tools by introducing the notion of a Transition Checking Matrix (TCM) which holds time and space constraints for model's transitions. By applying TS-U methodology, regular CTL formulae can be used without modifications to successfully check the system's model as in the traditional model checking based on the Kripke structure.

## 1 Introduction

Nowadays, with rapid advancements in communication technologies and miniaturizations of computational devices many of the current and future systems are, and will, become mobile. Computational and communication ability will be present in many devices from household items, office equipment, apart from the already present wearable devices such as mobile phone, PDA and the like. Thus, a ubiquitous computing environment will become a normal computing environment where various devices communicate in an ad-hoc way, sharing their physical resources and data. When designing applications in such a setting, it is important to check that the application performs as intended due to environment's dynamics, because devices in some area can appear, disappear and request services in a non-predictable random way. Therefore, in ubiquitous environment we should consider both space (room, building, neighborhood, etc.) and time, since different actions can be performed only in particular space at the particular time frame. On the same token, same actions can transition to some other space and continue or stop depending on the time, be it either discrete or in a periodic.

Thus, space and time are interrelated in the ubiquitous environment and should be used in parallel to check some application's correct behavior. In this paper we present a Temporal-Spatial methodology for system checking in the ubiquitous environment called TS-U, which can be used to model some specific ubiquitous system

(application) and then use the model checking procedures to reason and check for the correct behavior of a modeled system.

This paper is organized as follows. In Section 2 background research is presented where temporal and spatial logic is briefly explained, along with other researches on the temporal-spatial logic. Section 3 introduces the TS-U methodology where it is being applied on the example presented in the Section 4. Section 5 concludes the paper where future work is presented.

## 2 Background Research

Model checking based techniques [1, 2, 3] based on temporal logics [4] are used to express system requirements to verify the design of various systems, hardware or software. Two main paradigms are symbolic and explicit-state representations for the system states implemented in SMV [5] and SPIN [6]. In CTL model checking, the system is modeled as a Kripke structure and its desired properties to be verified are represented in CTL formulae. Kripke structure  $K = (S, T, L)$  is a labeled finite state machine, where  $S$  is a set of states,  $T$  is the transition relation which is required to be total and  $L$  is a labeling function, labeling transitions with atomic propositions  $A$  from one state to another. CTL formulas are concatenation of atomic propositions and basic operators such as “ $\sim$ ” standing for “not”, “ $\vee$ ” standing for “or” and characters “E”, “X”, “G” and “U” standing for “there exists”, “next”, “global” and “until”. In model checking using CTL, one would design a Kripke structure representing the system and label the transitions. In each state certain propositions would hold or not. Then, the system would be checked using a CTL formula in order to determine its validity. For example, if one would model a microwave-oven system, where the door can be opened or not and the power can be on or off, one could use the CTL operators to check that there is no case when the door is opened and power is on where microwave oven is radiating causing a safety hazard.

However, model checking using Kripke structure is not well suitable for continuously changing influences, but is more suited for discrete changes. For example, one could not model easily some variable whose influence depends on some interval that can be used to verify the validity of the system. For example, suppose a variable  $x$  can have a value from between 0 and 10, and depending on its value a system would have different properties that are valid or not. In such a case, a designer would have to introduce 10 different states to correctly model the system using a Kripke structure or a designer would have to select some degree of granularity, say introducing a state for which  $x$  value is between 0-5 and another for values between 6-10 to represent some interested states of the system; however the model may not be exact representation of the real world. Therefore, large number of states is required to model a system correctly causing a state explosion problem.

In this paper, proposed technique does not require additional states since restrictions are imposed on the transitions and not on states, thus existing tools can be used directly with minor modifications and variables based on intervals (i.e. time) can be successfully modeled.

In [7] a spatial logic based on regions and connections is presented where interval logic for reasoning about space is presented. Authors define a basic primitive relation

$C(x,y)$  that reads “ $x$  connects with  $y$ ” defined on regions and it holds whenever regions  $x$  and  $y$  share a common point. In terms of space, a subsumption hierarchy is defined on how two spaces can be related to each others consisting of 8 different relations: PO (partial overlap), TPP (tangential proper part), NTPP (non-tangential proper part), = (equal),  $TPP^{-1}$  (inverse of TPP),  $NTPP^{-1}$  (inverse of NTPP), EC (externally connected) and DC (disconnected). Thus, the logic is more commonly known as RCC-8. However, the RCC-8 was developed to be universal and general, whereas it can be much more simplified when the ubiquitous environment is considered, as in this paper. For example, in the RCC-8 logic there is a clear distinction between TPP (a,b) where  $a$  is enclosed inside  $b$  touching  $b$  in one point (tangential) and NTPP (a,b) where  $a$  is enclosed in  $b$  but not touching its boundary. However, if we suppose a classroom is inside a building, where both classroom and building have some shared and unique properties that have to be verified, we would not be concerned if a classroom is at the buildings corner (TPP) or somewhere not touching external building’s walls (NTPP). Thus, in this paper more simplified spatial relationships are addressed without the loss of generality when ubiquitous environment is considered.

In [8] a modal logic for mobile ambients is proposed whose purpose is to study mobility defined as a change of spatial configurations over time. Thus, logic proposed talks about space and time and has the Ambient Calculus as a model. Properties in such logic can hold at particular locations, i.e. space, and spatial configurations evolve over time as a consequence of the activities of some processes. Logic is mostly developed to automate the checking of mobile code in mobile environments for security purposes. However authors, when dealing with time and space issue, are concerned about the “sometime” and “somewhere” constraints. That is, the logic proposed is not concerned with specific time intervals or space, which is required in a ubiquitous environment. In other words, having “sometime” and “somewhere” constraints does not allow for a more complete and explicit model to be created whose properties can be checked. In this paper, specific time intervals and spatial relations are given, thus making modeling more expressive and correct in representing real world applications.

### 3 TS-U Methodology

TS-U methodology concerns time and space and applies it to the already existing model checking procedures as described in the previous section. Thus, we will deal with time and space issue separately in the following subsections and then present a method to combine them and apply them to the model checking useful in the ubiquitous environment.

#### 3.1 Time

Considering time and applying it to the ubiquitous environment we can consider four possible cases with or without repetition. In this paper, we adopt a 24-hour time system for clarity. The time can be one discrete time point, it can signify the time before or after some time point or it can specify a time period from one time point to

another time point and time can be represented as infinity, which means some event can occur at any time. Lastly, some time points can repeat themselves over a period of time as well. To clarify, let’s consider the following examples:

- “A meeting at 10, Tuesday, March 8<sup>th</sup>, 2005” is considered to be a discrete non-repetitive time point.
- “A meeting every Monday at 14” is considered a repetitive time point which repeats every 7 days.
- “Watching TV after 20 on Tuesday, March 15<sup>th</sup>, 2005 ” is considered a non-repetitive time point that occurs after 20 hours on March 15<sup>th</sup>, 2005.
- “A presentation from 15-16, on Tuesday, March 15<sup>th</sup>, 2005” is a time period which is a non-repetitive time period occurring only on March 15<sup>th</sup>, 2005 from 15 to 16 hours.
- “A class from 12-13 every Friday” is a repetitive time-period occurring from 12-13 hours every Friday.

To reduce the number of variables to represent time, we can adopt a time system like UNIX system time [9] which specifies the number of seconds elapsed since the beginning of the epoch (1/1/1970). Thus, instead of dealing with hours, minutes, seconds, days, months and years separately, we obtain one unique number for some specific time.

Adopting such a time representation, we can define possible time configurations more formally as in the Table 1 below:

**Table 1.** Possible time configurations

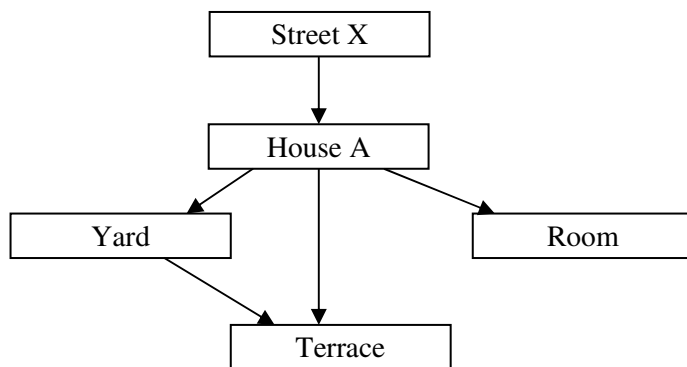
Time Category	Description
$\infty$	Infinite time
$t+\Delta r$	Discrete time point occurring once at time $t$ if $\Delta r=0$ . In case of repetition, $\Delta r$ represents the period of next occurrence from the starting time point $t$ .
$<(t+\Delta r)$	Represents time before some discrete time point $t$ when $\Delta r=0$ . In case of repetition, $\Delta r$ represents the period of the next occurrence of such time specification.
$>(t+\Delta r)$	Represents time after some discrete time point $t$ when $\Delta r=0$ . In case of repetition, $\Delta r$ represents the period of the next occurrence of such time specification.
$(t+\Delta r,d)$	Represents a time period starting at a discrete time point $t$ with duration of $d$ . In case of repetition, $\Delta r$ represents the period of the next occurrence of such time specification.

Since we are concerned about if certain transition in the Kripke structure can occur or not at some specific time compared with the allowed interval, it is not necessary to consider the relationships between intervals. That is, it is not relevant to check if some interval happens before or after some other time interval, since main concern is to check if the current system time matches some specified time interval or not to allow or not allow the transition to occur.

### 3.2 Space

Space is defined as a closed, bounded area where an activity can occur. Usually, as explained in the background section above, there are several possible cases how space can be configured, for example two space regions can touch each other tangentially, overlap or one can be enclosed in another, etc. However, in case of ubiquitous environment we are considering the space configuration at a specific time point. Thus, not all cases presented in [7] need to be considered. For example, in case two spaces touch each other tangentially, a person can be only in one space at the time if they are adjunct to each other (e.g. rooms next to each other). In case two spaces overlap or one space is contained in another and a person is in such a place, then a person is present in both places at the same time (e.g. room in a house; a person is both in the room and in the house at the same time), or in case person is outside the contained space (e.g. front door of the house). In case two spaces partially overlap each other, person can be in one space or the other, or in the overlapping area (e.g. house has a yard that has a terrace, however terrace is built on the yard; thus, person can be either in the house, on the yard or on the terrace which is both a house and a yard). Further, some space can have more actions that are allowed in it than the other. For example, a professor in a campus building has more resources and rights in his private office than on the hallway of the same building.

Therefore, a taxonomy of spaces as in [7] should be considered fully, and in the proposed approach the children would inherit the rules of what activities can occur (at a specific time) from their parents and can add additional rules to itself. An example of such taxonomy is depict in the Figure 1 below:



**Fig. 1.** An example taxonomy of space configurations and their relationships

Therefore, rules of the “*Street X*” that can occur there are inherited by the “*House A*”, since “*House A*” belongs to the “*Street X*”. Rules of both “*Street X*” and “*House A*” are inherited by both the “*Yard*” and the “*Room*” since they both belong to the “*House A*” and “*Street X*”, however they can add their own rules, if there such a need. Lastly, the “*Terrace*” inherits rules of both the “*Yard*” and the “*House A*” (and “*Street X*”), since it is both in a “*Yard*” and a part of the “*House A*”.

More formally we can define an inherit operator  $\leftarrow$ , for example  $A \leftarrow X$  that specifies that some space inherits from another space. That is, rules along with their specified time-frames are inherited from space  $X$  and now belong to the space  $A$ , since space  $A$  subsumes  $X$ .

As mentioned earlier, each space can have a rule set called RS. For the rule specification, we adopt propositional logic where the following Boolean operators are allowed: “ $\neg$ ” for NOT, “ $\wedge$ ” for AND and “ $\vee$ ” for OR. We write  $A[RS_A]$  meaning that a space  $A$  has a rule set  $RS_A$ . In case of inheritance, inherited rules from the parents are joined in the set of the children. For example,  $A[RS_A \cup RS_X] \leftarrow X[RS_X]$  means, that space  $X$  has a rule set  $RS_X$  and space  $A$  has both its own rule set  $RS_A$  and inherited rule set  $RS_X$ . Rules  $R$  in a set form a tuple of the propositions  $P_i$  and time specifications  $T_i$ , which is written as  $(P_i, T_i)$ . Thus,  $RS_X$  for the “*Street X*” can have an element such as  $(\text{walk}, \infty)$  where  $P_1 = \text{walk}$  and  $T_1 = \infty$ , meaning that someone is allowed to walk on the street at any time. However, sometimes inherited rules violate the policies that could be present in the children’s space, in which case they can be omitted from inheritance by the system modeler.

Therefore, considering the case as in Figure 1, possible rules, with inheritance displayed, could be as follows:

- (1)  $X[(\text{walk}, \infty)]$
- (2)  $A[(\text{sleep}, >100+500), (\neg\text{smoke}, \infty), (\text{walk}, \infty)] \leftarrow X[(\text{walk}, \infty)]$
- (3)  $\text{Yard}[(\text{smoke}, \infty), (\text{walk}, \infty)] \leftarrow A[(\text{sleep}, >100+500), (\neg\text{smoke}, \infty), (\text{walk}, \infty)] \leftarrow X[(\text{walk}, \infty)]$

In the above rule (1), a person is allowed to walk on the “*Street X*” anytime. Rule (2) says a person in the “*House A*” can sleep after time 100 with the repetition period of 500 (we omitted real time since epoch to keep the numbers small), he/she can never smoke and can always walk. Rule (3) states a person can always smoke and walk in the “*Yard*”, sleeping and not smoking rule is not inherited from the parent “*House A*” (suppose sleeping is dangerous and smoking is allowed so the non-smoking rule is overridden).

### 3.3 Transition Checking Matrix (TCM)

Once the treatment of time and space has been defined separately, for model checking methodology in the ubiquitous environment time and space has to be treated simultaneously, because of their interrelated relationship. In other words, some action in the ubiquitous application can be performed if and only if both time and space agree according to the rules. Thus, to check if some action is allowed or not we introduce the Transition Checking Matrix (TCM). TCM contains the action itself, space set and rule set, similar to the rules in the previous subsection. When model checking the application, TCM is consulted and if both space and time agree, an

action can be performed. TCM has the following structure, where example rows have been inserted from the previous subsection for clarity:

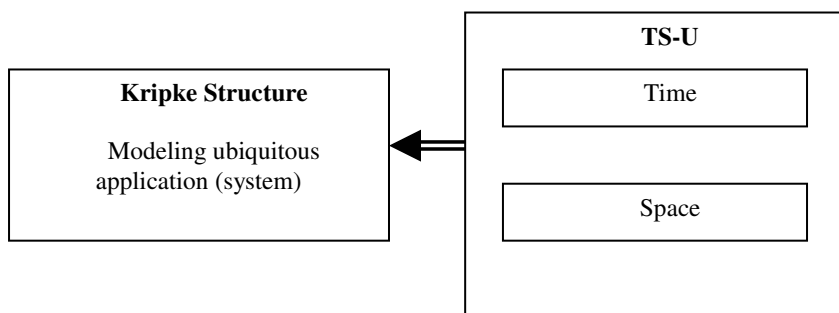
**Table 2.** Example of the TCM

Action	Space	Time
walk	X	$\infty$
sleep	A	$>100+500$
$\neg$ smoke	A	$\infty$
smoke	Yard	$\infty$

If some space has multiple actions, multiple rows are inserted with the action name and time constraint, as is the case for space A in the Table 2 above.

### 3.4 Model Checking Using TCM

Existing model checking explained in Section 2 of this paper, uses Kripke structure to model and check the system for correct behavior using rules specified by the CTL logic. In this paper TS-U is proposed that extend already existing model checking methodology, by applying TCM matrix for applying additional constraints on the transition labels. In other words, a transition in Kripke structure can occur from one state to another if both space and time match. Transition labels (actions) are put in the first column of the TCM matrix. Space and time are dealt as external variables that provide environment input to the ubiquitous system modeled with the Kripke structure. Conceptually, the system can be looked as depict in Figure 2 below:



**Fig. 2.** TS-U conceptual view

In Figure 2 above, on the left is the regular existing model using the Kripke Structure that models some ubiquitous system. On the right is the proposed TS-U methodology that provides external input concerning time and space to the Kripke structure. Model checker uses CTL to check the system. However, each time a transition should be made TS-U is consulted where transition is checked against TCM concerning space and time and thus a transition is either allowed, if both time and

space constraints are matched for such a transition, or not allowed if either time or space or both are not matched.

It is important to note that during model checking, a time frame is specified in order to prevent infinite execution (say, model checker is in a certain state and no transition can occur since space/time constraint is never met). For example, if we create a model of some ubiquitous application in a university environment, say “smart campus” studied in [10], we may be interested in checking that the system operates as expected in a one-week time frame, since during the semester events are usually repetitive. Thus, granularity of time should be a variable parameter as a part of model checker program’s input. On the same token, we assume location is known *a priori* every time verification is performed against TCM. Thus, when the time frame period finished model checking stops and CTL formula results are presented. If for some CTL formula model checker did not finish its evaluation and the time frame period finished, such formula is evaluated to *false* since for the interested time period its truth is false. Indeed, for the same formula if the period is longer (say, instead of one week, two weeks) the same formula may be evaluated to true, if some action has repetition granularity of two weeks. However, as an input to the model checker when time was specified as “one week”, the formula would be evaluated to *false*.

#### 4 Example Application Scenario

In order to show the application scenario of proposed TS-U methodology, let’s consider a university environment, consisting of a professor as an entity performing ubiquitous actions, his office, a lecture room and a meeting room. A professor can perform various activities in his office, such as use email, phone or fax machine. He has a lecture every Monday from 14:00-15:00 where he can use a projector and whiteboard and a meeting every Tuesday from 10:00-11:00 where he can use a notebook computer. Activities performed are: “give lecture”, “return to office” and “go for a meeting”. Thus, the following Kripke structure can be devised as presented in the Figure 3 below:

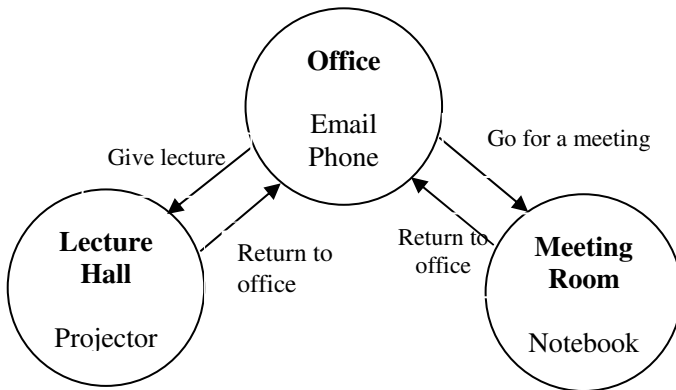


Fig. 3. Kripke structure of an example scenario



In Figure 3 above, regular Kripke structure is devised, where existing model checking techniques based on CTL can be used. However, in the ubiquitous environment, constraints on transitions (“give lecture”, “return to office” and “go for a meeting”) have to be made, since many of the resources are limited and can only be used by persons at a specific time and place. For example, in the scenario above, a professor can give lecture only from 14:00-15:00 on Monday where his role is of a lecturer, but he can attend some other lecture, say workshop, in the same lecture hall given by others, so he can not use a projector and a whiteboard at that time in that place. Thus, constraints based on proposed TS-U methodology can be as follows:

**Table 3.** TCM of TS-U for the example scenario

Action	Space	Time
Give Lecture	Lecture Hall	(100+3600,604800)
Return to office	Office	$\infty$
Go for a meeting	Meeting room	(300+3600,604800)

In the table above, a professor can transition from an office space to the lecture hall space at time 100 (suppose time 100 is Monday at 14:00) and use its resources for 1 hour with a repetition of 7 days. Naturally, he can return to his own office at any given time, even during the lecture or meeting, etc. Lastly, he can transition from his office to the meeting room and use its resources at a time 300 (suppose time 300 is Tuesday at 10:00) where he can use the resources for 1 hour with a repetition of 7 days. It should be noted, that TCM does not contain rows for a professor transitioning to the lecture hall in case of a workshop mentioned earlier, since no resources would be needed when transition occurs, thus validity of the model should not be in question.

Once the above model has been created using the proposed approach, existing CTL formulae can be used to verify the properties of interest without modifications, like they would be used if the model was created using a regular Kripke structure as in the previous approaches discussed in Section 2 of this paper.

## 5 Conclusion

In this paper an TS-U methodology was proposed that extends existing model checking techniques based on CTL logic, that combines both space and time which are additional constraints on the model being checked, which is an important factor in the ubiquitous environment. TS-U utilized Transition Checking Matrix (TCM) to verify if some transition can occur or not. For future work, we would like to extend existing model checking tool, SMV [5], to implement proposed TS-U methodology to obtain more concrete results and also to extend TS-U to another level where actions can be tied to individuals, thus same model in terms of the Kripke structure can be reused.

## Acknowledgement

Doo-Kwon Baik and Hoh Peter In are the co-corresponding authors.

## References

- [1] E. Clarke, E. Emerson, and A. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems*, 8(2):244-263, 1986.
- [2] J. Quielle and J. Sifakis. Specification and verification of concurrent systems in CESAR. In *Proceedings of the Fifth International Symposium in Programming, LNCS 137:337-351*, 1981.
- [3] Edmund M. Clarke, Jr., Orna Grumberg and Doron A. Peled. *Model Checking*. The MIT Press, Cambridge, Massachusetts, 1999.
- [4] A. Pnueli. The temporal logic of programs. In *Proceedings of the 18<sup>th</sup> Symposium on Foundations of Computer Science*, pages 46-57, 1977.
- [5] K. McMillan. *Symbolic model checking: an approach to the state explosion problem*. PhD thesis, School of Computer Science, Carnegie Mellon University, 1992.
- [6] E. Clarke, E. Emerson. Design and synthesis of synchronization skeletons using branching-time temporal logic. In *Proceedings of the Workshop on Logic of Programs, LNCS 131:52-71*, 1981.
- [7] D.A. Randell, Z.Cui, A.G.Cohn. A Spatial Logic based on Regions and Connection. In *Proceedings of the 3<sup>rd</sup> International Conference on Knowledge Representation and Reasoning*, Morgan Kaufman, 1992.
- [8] L. Cardelli, A.D. Gordon. Anytime, Anywhere Modal Logics for Mobile Ambients. In *Proceedings of POPL Conference*, 2000.
- [9] Bell Labs. *Unix Programmer's Manual*. First Edition, 1971.  
<http://cm.bell-labs.com/cm/cs/who/dmr/1stEdman.html>
- [10] L. Barkhuus, P. Dourish. Everyday Encounters with Context-Aware Computing in a Campus Environment. In *Proceedings of UbiComp 2004 conference on Ubiquitous Computing*, 2004.

# Ubiquitous Learning on Pocket SCORM

Hsuan-Pu Chang<sup>1</sup>, Wen-Chih Chang<sup>1</sup>, Yun-Long Sie<sup>1</sup>, Nigel H. Lin<sup>1</sup>,  
Chun-Hong Huang<sup>1</sup>, Timothy K. Shih<sup>1</sup>, and Qun Jin<sup>2</sup>

<sup>1</sup> Department of Computer Science and Information Engineering,  
Tamkang University, Taiwan, Republic of China  
sul@mail.mine.tku.edu.tw

<sup>2</sup> Department of Human Informatics and Cognitive Sciences,  
Waseda University, Tokorozawa-shi, Saitama 359-1192, Japan  
jin@waseda.jp

**Abstract.** With advanced technologies, computer devices have become smaller and powerful. As a result, many people enjoy ubiquitous learning using mobile devices such as Pocket PCs. Pocket PCs are easy to carry and use as a distance learning platform. In this paper, we focus on the issues of transferring the current PC based Sharable Content Object Reference Model (SCORM) to Pocket PC based. We will also introduce the Pocket SCORM Run-Time Environment (RTE) which has been developed in our lab. Pocket SCORM architecture is able to operate, even when the mobile device is off-line. It will keep the students' learning record. When it is on line, the records will then be sent back to Learning Management System (LMS). With memory limitation, we provides course caching algorithm to exchange the course content on the Pocket SCORM.

**Keywords:** Pocket PC, PDA, SCORM, Distance Education, Ubiquitous Learning.

## 1 Introduction

Because of the advantage in size, Pocket PC is easy to be carried on hand or in the pocket. This advantage makes it a suitable platform for distance education because E-Learners can keep studying the course materials while they are away from their desk. Without the limitations of time and space, E-Learners can read their learning materials while they are walking, taking bus, and whenever they have leisure time to turn on their Pocket PCs. This new learning style provides some extra learning time for E-Learners who live in this rushing world.

Although Pocket PC devices have been improved in both computing power and memory storage, they are still with lots of limitation compared with laptop or desktop computers. The following points are some differences between the Pocket PC devices and laptop or desktop computers.

### Connection Mechanism

Normally, E-Learners will be on-line when they are using laptop or desktop computers. Since E-Learners would stay at the same location longer when using their

laptop or desktop computers than using a Pocket PC devices. As a result, they will be kept connected with the LMS, which is running on the server, all the time. This is different from Pocket PC devices because these devices are designed to be portable.

### **Courseware Import and Export**

As we mentioned previously, Pocket PC devices sometimes could be disconnected from the network; however, we hope E-Learners are able to learn even when the network is not existed. In order to enable this functionality, we need to seek a way to allow the courseware to be temporary stored on the hand carried devices.

### **Learning Records Buffering**

For a distance education standard such as SCORM, it requires the LMS to be able to keep tracking on learners' learning records so these records could be used to determine learners' progress or maybe transfer to other LMS where learner continue his education. As a result, it is important for a SCORM compatible platform to be able to store learners' learning records.

In this paper, we proposed a platform which focuses on the pocket devices. There is also a LMS which will be SCORM compliant to support the Pocket SCORM platform and be the major data store. The paper organization is listed as following. In Section 2, we introduce some related works of SCORM and systems on the pocket device. Section 3 shows the architecture of our Pocket SCORM architecture. In Section 4, we introduce the Pocket SCORM RTE which has been developed in our lab. In Section 5, we introduce the course caching algorithm in Pocket SCORM. Finally the conclusion and the future works are discussed in Section 6.

## **2 Related Works**

Distance education enables E-Learners to learn without the restrictions of both time and space. There are many web-based courseware have been developed to allow learners to browse course content via a browser. SCORM (Sharable Content Object Reference Model) is a standard which is proposed by ADL (Advanced Distributed Learning) [1]. SCORM is aiming at the standardization of computer based teaching components. There are some papers related to SCORM have been published. Some advantages such as portability of learning content, the standardized communication interface between the LMS and WBTs, and supporting the reusability of learning object came along with this new proposed standard. However, there are some problems as well such as the market value of SCOs, the process of producing WBTs on the basis of different SCO providers, the maintenance of SCOs and WBTs, and the quality of WBTs based on SCOs of different providers. A review discussed these issues can be found in [2]. Another paper demonstrated the Implementation of Content Repository Management System is referenced in [3]. In "Using SOAP and .NET web service to build SCORM RTE and LMS" [4], the XML Web Service based LMS and RTE was introduced. There was another system developed for automating the generation of SCORM-Based multimedia product training manuals was introduced in [5].

Personal Digital Assistants (PDAs) have become new learning tools for distance education in recent years. Portability of the PDAs was welcomed by students, and

advantageous was advantageous, limitations such as the small screen size, navigation difficulties, and slow and error-prone methods for entering text, made it difficult to read and interact with document on the PDA [6]. Some students' experiences for reading course materials by PDAs were experimented. There are also some applications developed for educational purposes. TekPAC (Technical Electronic Knowledge Personal Assistant Capsule) was introduced in [7]. TekPAC was developed for providing access to readily available electronic information, allowing the user to perform tasks at locations with all schematics, photos, videos and BKMs readily available, and integrating key interventions to raise performance of target audience. PDAs have also been adopted in medical field as a tool for education. There were some PDA Projects at Virginia Commonwealth University being introduced in [8].

### 3 Pocket SCORM Architecture

In proposed architecture, we pointed out two types of connection for a Pocket PC to connect with LMS Server. One type is Pocket PC is connected to the server through wired or wireless network to the internet. The other is Pocket PC connects to the server via PC to the internet while Pocket PC is synchronizing with the PC.

In order to make courseware reusable, a standard representation of contents and structures must be enforced. The Content Aggregation Model (CAM) serves this purpose. CAM can be discussed in three parts: the Content Model, the Metadata, and the Content Packaging.

There are three major modules within the Pocket SCORM Architecture. In the following three sub-sections, we will show the details of each of them.

#### 3.1 Pocket SCORM Run-Time Environment

There are six major components which are included in Pocket SCORM Run-Time Environment. All these components work together to form the whole Pocket SCORM Run-Time Environment. They are listed as below:

##### **Communication Agent**

The Communication Agent is used when the pocket devices try to communicate with the SCORM LMS Server. When E-Learners try to download the SCORM based courseware from the LMS Server, it will receive the packed courseware and pass it to Data UnPacking Agent. If there are some learning records need to be sent back to the LMS server, the Communication Agent will connect to the LMS Server and send the packed learning records back to the server. We considered using Simple Object Access Protocol (SOAP) [9] to be our transmission protocol to make our services of server side more extendable.

##### **Data Packing Agent**

We implemented the Data Packing Agent to reduce network load. This agent will pack the data before sending it to Communication Agent.

##### **Data UnPacking Agent**

Package Interchange File (PIF) is the exchange file format. Therefore, we need Data UnPacking Agent to unpack the PIF file.

**Learning Agent**

The Learning Agent will keep tracking on the learner's learning records when the course is started. The learner might be off-line, the Learning Agent stores those learning records in the SCORM PDA Database temporarily. After it is on-line, those temporary stored learning records will be sent back to the Server.

**SCORM PDA Reader**

Due to the small screen restriction, normal web-based course content is not suitable for learners. They might need to use a stylus to control the scrollbars inside the Pocket Browser. It is inconvenient for the learners to use. To overcome this drawback, we designed "Reflow" function in the SCORM PDA Reader. "Reflow" function will adjust the content to make it fit in the width of the display width on our reader.

**SCORM PDA Database**

SCORM PDA Database has the capability of storing the temporary learners' learning records and the downloaded courseware.

**3.2 PC Dock**

If the Pocket PC without the ability to be on-line, it will require a PC Dock to be able to connect to the LMS server. There is a Synchronization Agent inside the PC Dock. The Synchronization Agent will perform the data transmission job between Communication Agent on the Pocket SCORM RTE and XML Web Service on the SCORM LMS Server.

**3.3 SCORM LMS Server**

There are two major components involved in SCORM LMS Server. One is the SCORM Data Repository, and the other is Pocket SCORM Service API. SCORM LMS Server provides distance education courseware which follows SCORM Data Model definition. The learners' information is also saved in the SCORM LMS Server. When a learner connects to the LMS Server, he or she needs to first logon the system before he or she can access any course materials. The usage of these two major components is stated as below:

**SCORM Data Repository**

The SCORM Data Repository stores all the course materials which follow SCORM Data Object Model. In the paper, we mainly focused on the Pocket PC devices. Therefore, we only care about how SCORM Data Repository interacts with our Pocket SCORM Service API. Nevertheless, this data repository should also support any SCORM based API. Furthermore, learners' learning records are also stored in this data repository. These SCORM based learning records should also be able to interact by using either Pocket SCORM Service API or any SCORM based API.

**Pocket SCORM Service API**

Ideally, Pocket SCORM Service APIs should be same as normal SCORM based APIs. We hope defined Pocket SCORM Service APIs can be widely applied by other applications. We tried to build Pocket SCORM Server APIs by adopting XML Web

Service [10] technology. XML Web Service takes SOAP as its transmission protocol. One of the advantages of using XML Web Service as our APIs is the accessibility. Since SOAP is loosely coupled protocol by using XML wrapped envelope to invoke APIs, this vantage makes XML Service APIs can be accessed by any platform which follows SOAP protocol to acquire the service. As a result, we hope to implement our Pocket SCORM Service APIs as XML Web Services.

#### 4 Pocket SCORM RTE Implementation

Based on the proposed architecture, we have been working on the implementation of the whole architecture. Up to present, we have completed some portion of the whole Pocket SCORM Architecture. There were some components which comprises the Pocket SCORM Run-Time Environment. Some user interfaces and functions will be introduced in this section.

In fig. 1, there are two user interfaces. On the left hand side, it shows the UI when student try to logon to Pocket SCORM RTE. Because we need to track on learners' learning records, the learner needs to provide his identity before studying the course materials. There is also an important issue need to be taken care. We need to make sure is the same user who is studying the courseware. In order to make sure learner's identity, the logon UI will be pop-up each time when Pocket PC has been turned off and turned back on again. On the right hand side of fig. 1, it shows a list of imsmanifest files which represent the each different course structure.

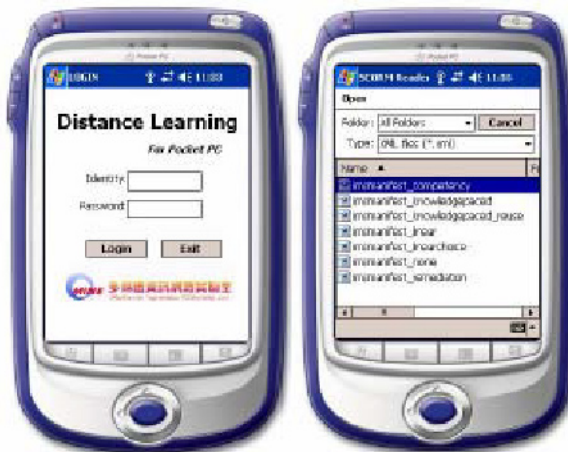


Fig. 1. Logon Interface and Course List View

In SCORM there are two major structures of a course were defined. One of them is knowledge based course structure, which is shown as on the left hand side of fig. 2, and the other one is linear based structure, which is shown as on the right hand side of fig. 2. Our Pocket SCORM RTE is capable to display the course structure according to defined imsmanifest file which learner chose to load.



Fig. 2. Knowledge based and Linear based Structure

Fig. 3 has shown our SCORM PDA Reader and a glance of our SCORM PDA Database. On the left hand side of fig. 3, it shows the UI of SCORM PDA Reader. As we introduced in section 3, our SCORM PDA Reader provides “Reflow” function to enhance the ease of reading with one hand only. In order to gain better performance, we adopted SQL CE which is a compact version of Microsoft SQL Server. On the right hand side of fig. 3, we have shown the database content within SQL CE database.



Fig. 3. SCORM PDA Reader and SCORM PDA Database



## 5 Pocket SCORM Course Caching Algorithm

A SCORM course can be divided into several parts called clusters. A cluster includes a node of the learning content and all immediate children of the node. Every cluster has its own content size, and it is also a base unit to be loaded and to be replaced. A course developer can set the sequencing rules for each cluster, and those rules only affect the learning order of the cluster.

### 5.1 Cluster Download Order

Account to the setting of sequencing rules, it is available to forecast that what clusters the learner may read next. Loading clusters by a better order, it can increase the hit ratio of caching. With the increasing of hit ratio, a learner do not need to download new clusters so frequently. In order to find the learning order, some of the factors are shown in the following.

**Control Mode:** The setting of sequencing rule. Because the setting will affect the learner's learning order, it is necessary to check the value of rules. If the "flow" flag is true, learner will be forced to read the first child node in a cluster. If the "choice" flag is true, learner can choice any child node in a cluster.

**Size:** The content size of each cluster. With a fix size of storage, it can contain more clusters by loading small clusters first. As the number of clusters increase the hit ratio raised, and the learner accesses the downloaded cluster more properly.

Summing up these factors can forecast the order with the following steps.

PC: Parent Cluster, IC: Immediate Child Cluster of PC,  
Capacity: Capacity of storage  
Input: Capacity, PC Output: Cluster Download Order

```
Function DownloadOrder(Capacity, PC){
  If(Flow Control Mode of PC is true){
    Each IC until Capacity is not enough{
      Output << Cluster Number of IC;
      Capacity = Capacity- Size of the IC;
      DownloadOrder(Capacity, IC);
    }
  }
  Else If(Choice Control Mode of PC is true){
    Insert IC into ChildArray non-descending until
      Capacity is not enough or no more IC;
    Output << Cluster Number of the ICs in ChildArray;
    Capacity = Capacity - Size of the ICs in
      ChildArray;
    Sort the ICs in ChildArray according to
      their Cluster Number;
    Each IC in ChildArray{
      DownloadOrder(Capacity, IC);
    }
  }
}
```

## 5.2 Cluster Replacement Algorithm

When using a storage which is smaller than the whole course, it is inevitable to replace some clusters with the new loaded ones. In order to compute what clusters should be dropped first, we assume that there has a factor called “Distance” which can determine what cluster is useless by some parameters like cluster size, download time, and so on. With the factor, it can describe the replacement as replacement algorithm.

RC: Removed Cluster, DC: Desired Cluster,  
 PC: Parent Cluster,  $\alpha$ : Threshold Value  
 Input: DC, Output: Dropped Cluster Number

```
Function Replacement(DC){
  While(Capacity <=  $\alpha$ ){
    RC = the clusterj have the max Distancei,j;
    // i: DC, j: each cluster in the storage
    RC removed from the storage;
    Output << Cluster Number of RC;
    Capacity = Capacity + Size of the RC;
  }
  PC = DC;
  While(Capacity > Size of the PC){
    Download PC;
    Download DownloadOrder(Capacity, PC);
    PC = Parent cluster of PC;
  }
}
```

Downloading a course in good order can make the reading more smoothly and needs less replacement. With a faultless replacement algorithm, the new downloading clusters just replace the little significant ones. The two functions of the caching algorithm working together ensure that a learner does not need to reconnect to network and re-download some clusters again and again.

## 6 Conclusion and Future Works

Distance Education provides a convenient and flexible learning environment. Various kinds of distance learning methods have broken the time and space limitation. It is possible for people to learn anytime anywhere. To extend this flexibility and to make E-Learners able to learn at any location, our proposed Pocket SCORM architecture in this paper makes the dream come true. The Implemented Pocket SCORM RTE components were also introduced. We hope our proposed architecture can make E-Learners to learn easier by using a pocket device which can be carried to anywhere and enable E-Learners to learn anytime.

## References

- [1] Advanced Distributed Learning (ADL) (2003), <http://www.adlnet.org>.
- [2] Oliver bohl, Dr. Jorg Schellhase, Ruth Sengler, and Prof. Dr. Udo Winand (2002), “The Sharable Content Object Reference Model (SCORM) – A Critical Review”, Proceedings of the International Conference on Computers in Education (ICCE’02)
- [3] Jin-Tan David Yang, and Chun-Yen Tsai (2003), “An Implementation of SCORM-compliant Learning Content Management System – Content Repository Management System”, Proceedings of the The 3rd IEEE International Conference on Advanced Learning Technologies (ICALT’03)
- [4] Timothy K. Shih, Wen-Chih Chang, Nigel H. Lin, Louis H. Lin, Hun-Hui Hsu, and Ching-Tang Hsieh (2003), “Using SOAP and .NET Web Service to Build SCORM RTE and LMS”, Proceedings of the 17th International Conference on Advanced Information Networking and Applications (AINA’03)
- [5] Peiya Liu, Liang H. Hsu, and Amit Chakraborty (2002), “Towards Automating the Generation of SCORM-Based Multimedia Product Training Manuals”, Proceedings of 2002 IEEE International Conference Multimedia and Expo, 2002. (ICME '02)
- [6] J. Waycott, and A. Kukulska-Hulme (2003), “Students’ experiences with PDAs for reading course materials” Personal and Ubiquitous Computing Volume 7, Issue 1 (May 2003), ISSN:1617-4909
- [7] Catherine Weissenborn, and Frank J. Sanchez (2001), “TekPAC (Technical Electronic Knowledge Personal Assistant Capsule)”, 2001 IEEE International Semiconductor Manufacturing Symposium
- [8] Kay Sommers, Jane Hesler, and Jim Bostick (2001), “Little Guys Make a Big Splash: PDA Projects at Virginia Commonwealth University”, Proceedings of the 29th annual ACM SIGUCCS conference on User services (SIGUCCS’ 01)
- [9] World Wide Web Consortium (W3C) (2003), <http://www.w3.org/TR/soap12-part0>
- [10] World Wide Web Consortium (W3C) (2003), <http://www.w3.org/TR/ws-arch>

# An Application Based on Spatial-Relationship to Basketball Defensive Strategies

Su-Li Chin<sup>1</sup>, Chun-Hong Huang<sup>2</sup>, Chia-Tong Tang<sup>2</sup>, and Jason C. Hung<sup>3</sup>

<sup>1</sup> Dept. of Physical Education, Tamkang University,  
Tamsui, Taipei Hsien, Taiwan 251, R.O.C.  
csunny@mail.tku.edu.tw

<sup>2</sup> Dept. of Computer Science and Information Engineering,  
Tamkang University, Tamsui, Taipei Hsien, Taiwan 251, R.O.C.  
690190185@s90.tku.edu.tw

<sup>3</sup> Department of Information Management,  
Northern Taiwan Institute of Science and Technology,  
Peitou, Taipei, Taiwan 112, R.O.C.  
jhung@cs.tku.edu.tw

**Abstract.** This paper aims to develop a simulated system used for teaching and training basketball defensive strategies. Respectively, defensive strategies can be described within one method by editing video recorded from basketball games into desired clips for analysis and storing them into the database. In this paper, we used Spatial-Temporal Relationships to describe the local defensive movements by the basketball players in a game. The system will automatically capture tracks of defensive movements by the basketball players in the video clips, from which basketball coaches and players can learn various defensive strategies within the shortest period of time. The simulated system is expected to become a computerized educational aid to basketball teaching and training and to replace the unscientific and stereotyped system of basketball teaching and training.

## 1 Introduction

Basketball is an open sport. In this paper, we aim to develop a simulated system used for teaching and training basketball defensive strategies [1]. No matter whether on defense or offense, basketball players have to react according to their opponent's movements [2]. The success of a team depends on the degree of teamwork. A coach must have professional knowledge of basketball and he or she directly tells the players the training topics, from which the players can learn the key to successful defense. Therefore, the coach's pre-training preparation in collecting as well as sorting the information concerning the opponent teams, and how to oppose each tactic, plays an extremely important role in the field of basketball [3].

When it comes to basketball tactics, what we basically understand are no more than concepts of space, ball, and players (offensive as well as defensive). How to move? When to dribble? And when to pass the ball to teammates? If we can utilize a

computer assisted teaching module, with theories of basketball tactics installed in the program, we are confident that the establishment of a simulated system concerning basketball tactics will facilitate the coach's preparation work. Figure 1 shows the overview of the system.

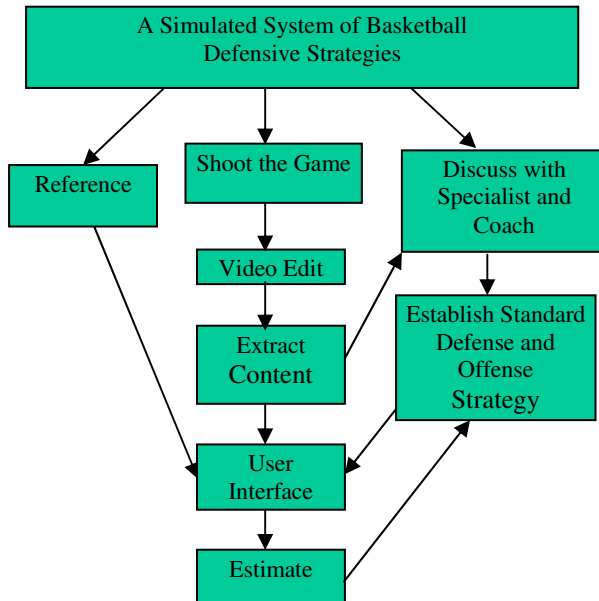


Fig. 1. The overview of system

## 2 Background

People in love with basketball would definitely know that a tactics-board is a white board on which marker pens or colored magnets are used to demonstrate specific tactics. Oftentimes, we see coaches draw the route or move the colored magnets to inform the players of the tactic message. Coaches have to express the tactics and draw the routes or move the colored magnets at the same time. The situation will go like this: Now, Position 1 dribbles the ball to the baseline, and meanwhile, Position 2 should..., and Position 3 needs to... Movements of the five players change accordingly, but coaches cannot draw five routes or move five magnets at the same time. Besides, the speed of each player and the position after the movement cannot be clearly displayed in terms of the relative space among the player, the teammates, and the ball. Those players who have a tacit understanding of the coaches' directions may quickly reach an agreement with the coaches; however, for newcomers or players who need time to accustom themselves to the situation, it would be totally different from the former.

The physical strength of a player is limited. Players are capable of experiencing a five to eight-hour training days. In addition to the tiresome training, a method with

scientific basis to improve the players' abilities is needed. Nations with strong athletic programs such as the United States, Russia, and China all invest huge amounts of money in scientifically researching the most effective method to improve sports performance. However, the scientific research concerning psychology, physiology, or movement analysis, from either magazines or The Discovery Channel, emphasizes the importance of strengthening personal quality and sports performance. It is strange that, when compared to players of other Asian countries, Taiwanese basketball players have comparable strength, skills, conception, training duration, and training intensity. However, players seem not to be able to optimize their potential. The underlying factor behind this lies in the success rate of teamwork coordination and tactical execution.

Therefore, the purpose of the present study aims to establish a simulated system used for teaching and training basketball defensive tactics. With the facilitation of the system, the players not only have a more profound understanding of the tactics but also maintain a clearer concept in executing the tactics, without the coaches' repeating explanations.

The remainder of the paper is organized as follows. The next Section presents the method for capturing the moving objects and defining the spatial relationship. Section 4 describes the Experimentation and Result. Finally conclusions and future work are drawn in Section 5.

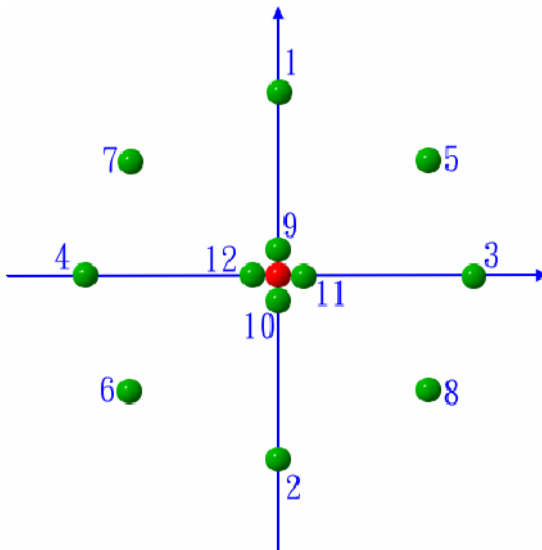
### **3 Capture the Moving Objects and Define the Spatial Relationships**

Tracking objects in an image sequence has been discussed in many papers [4] [5] [6]. The method we used to track objects is similar as in [7]. However, [7] treats two or more objects as one object when they may move too close to each other. In our system, we discriminate objects as individuals, and use the colors of sportswear to distinguish one team from the other. Then, we extract the trajectories and movements of the players from the video which is recorded from an overhead view as shown in figure 2. The purpose of doing so is to avoid the heavy collision of players brushing past one another. In analyzing a sequence of players, players are represented by using silhouette images. In this paper, we used Spatial-Temporal Relationships [8] to describe the local defensive movements by the basketball players in a game, since each silhouette image needs to be assigned a unique number initially, as it will help us to conveniently identify the spatial relationship between each object. According to figure 3, we can define the 12 spatial relationships between each defensive player. The spatial relationship can be appropriately applied to basketball defensive strategy. Then, we reconstruct a spatial relationships table which represents a unique ID number for each spatial relationship as shown in table 1.

Here we only consider 12 spatial relationships. We do not consider the relationship for example: "A is up right side of B and close to B" due to object A and object B are too close and are the team partner. In this paper, spatial relationships are used to evaluate defensive strategies such as "2-3 local defensive", "3-2 local



**Fig. 2.** To Film the basketball game with an overhead view



**Fig. 3.** The distribution of 12 spatial relationships

defensive” or “2-1-2 local defensive”. Figure 4 shows the topologies of these defensive strategies and they would be the standard defensive strategies which are stored in the database. In figure 5, there exist six objects A、B、C、D、E and F. A-E are players and F is the basketball stand which plays a role as benchmark. Generally, the topology for a defensive strategy does not vary dramatically in an image sequence, since a team enforces a defensive strategy with certainty. Different relationships have their own ID number and the relationship sets can be represented by the matrix for each frame, since different defensive strategies have different spatial

relationships. As shown as the topology in figure 5, the spatial relationships can be represented by the 6X6 SP matrix as follow:

$$SP_i^j = \begin{matrix} & A & B & C & D & E & F \\ \begin{matrix} A \\ B \\ C \\ D \\ E \\ F \end{matrix} & \begin{matrix} 0 & 4 & 5 & 7 & 1 & 7 \\ 0 & 0 & 5 & 7 & 5 & 1 \\ 0 & 0 & 0 & 4 & 7 & 7 \\ 0 & 0 & 0 & 0 & 5 & 5 \\ 0 & 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{matrix} \end{matrix}$$

This matrix represents the spatial relationship for the  $i^{th}$  frame of video clip  $j$ . For our system, we have  $n$  SP matrix, since we choose  $n$  frames from every chip equally. The set of SP matrix can be represented as follows.

$$SP^j = \{SP_1, SP_2, \dots, SP_i, \dots, SP_n\}$$

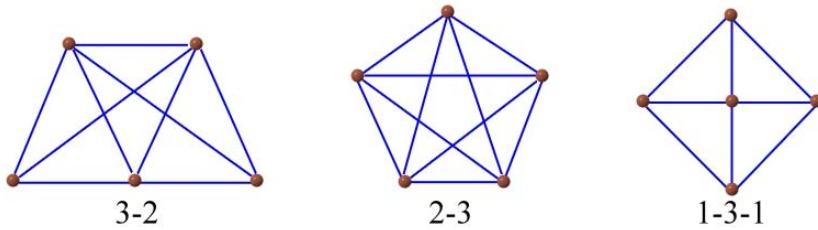


Fig. 4. Three topologies of defensive strategy

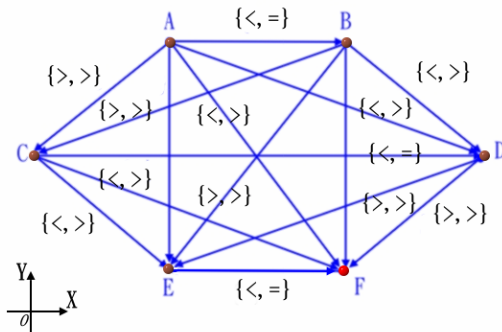


Fig. 5. The topology of defensive based on spatial relationship



We could calculate the similarity among different defensive clips. The distance *dist* between *SP* matrixes of each frames of different clip is obtained according to table 2.

$$dist_{(i)} = SP_i^j \Leftrightarrow SP_i^k \quad 1 \leq i \leq n \tag{1}$$

$SP_i^j$  : The spatial matrix of  $i^{th}$  frame of clip  $j$

$SP_i^k$  : The spatial matrix of  $i^{th}$  frame of clip  $k$

For example, if

$$SP_i^k = \begin{array}{c} \begin{array}{cccccc} A & B & C & D & E & F \end{array} \\ \left| \begin{array}{cccccc} 0 & 4 & 5 & 2 & 1 & 10 \\ 0 & 0 & 5 & 7 & 5 & 1 \\ 0 & 0 & 0 & 4 & 7 & 2 \\ 0 & 0 & 0 & 0 & 5 & 5 \\ 0 & 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right| \begin{array}{l} A \\ B \\ C \\ D \\ E \\ F \end{array} \end{array}$$

Then *dist*(*i*) between  $SP_i^j$  and  $SP_i^k$  is

$$\begin{aligned} dist_{(i)} &= (0+0+9+0+4) \\ &\quad + (0+0+0+0) \\ &\quad + (0+0+9) \\ &\quad + (0+0) \\ &\quad + (0) = 22 \end{aligned}$$

**Table 1.** 12 spatial relationships

ID	Relationships	Judgments(X, Y)
1	A is on the top of B	(=, >)
2	A is under of B	(=, <)
3	A is right side of B	(>, =)
4	A is left side of B	(<, =)
5	A is up right side of B	(>, >)
6	A is up left side of B	(<, <)
7	A is bottom left side of B	(<, >)
8	A is bottom right side of B	(>, <)
9	A is on the top of B and close to B	(=, m)
10	A is under of B and close to B	(=, mi)
11	A is right side of B and close to B	(mi, =)
12	A is left side of B and close to B	(m, =)

**Table 2.** The distance between each spatial relationships

ID	1	2	3	4	5	6	7	8	9	10	11	12
1	0	6	6	6	3	9	3	9	5	1	5	5
2	6	0	6	6	9	3	9	3	1	5	5	5
3	6	6	0	6	3	9	9	3	5	5	1	5
4	6	6	6	0	9	3	3	9	5	5	5	1
5	3	9	3	9	0	12	6	6	8	4	4	8
6	9	3	9	3	12	0	6	6	4	8	8	4
7	3	9	9	3	6	6	0	12	8	4	8	4
8	9	3	3	9	6	6	12	0	8	4	4	8
9	5	1	5	5	8	4	8	8	0	4	4	4
10	1	5	5	5	4	8	4	4	4	0	4	4
11	5	5	1	5	4	8	8	4	4	4	0	4
12	5	5	5	1	8	4	4	8	4	4	4	0

And the similarity *SoD*(*Similarity of Defensive*) between two defensive clips *j* and *k* is shown as followed:

$$SoD = \frac{1}{\frac{1}{n} \sum_{i=1}^n dist_{(i)}} = \frac{1}{\frac{1}{n} \sum_{i=1}^n (SP_i^j \Leftrightarrow SP_i^k)} \tag{2}$$

According the value of *SoD*, we could find similar defensive strategies in the database. The system supports a GUI to display the active similar defensive shot. This mechanism helps coaches to find the standard defensive technique for teaching and they could learn the usage frequency of the defensive strategy by the opponent.

### 4 Experimentation and Result

In our system, we need camera installation and proper clip editing, since we will just evaluate the defensive strategies. We should pre-edit the video and cut out the suitable clips that we want. The average time period of each clip is 20 seconds. However, the number of frames is probably different among clips which would impede comparison between defensive strategies. To solve this problem, we should choose enough average frames to make sure each clip would have an equal or close

on number of frames. The figure shows the GUI and the results of a query. We experimented with a desktop PC of Pentium-4 3.0 GHz. In this system, we marked the goals first before we extracted the locations of the players as shown in the video in the figure 6. The upper right side of figure 6 shows the defensive location of the players. After extracting the locations, the system would record the spatial relationships of every frame in the database. And we can query for the similarity of defensive strategies from the database. Presently, our database has 361 specimens. We still collect and film basketball games for expanding the number of specimens to be stored in the database.

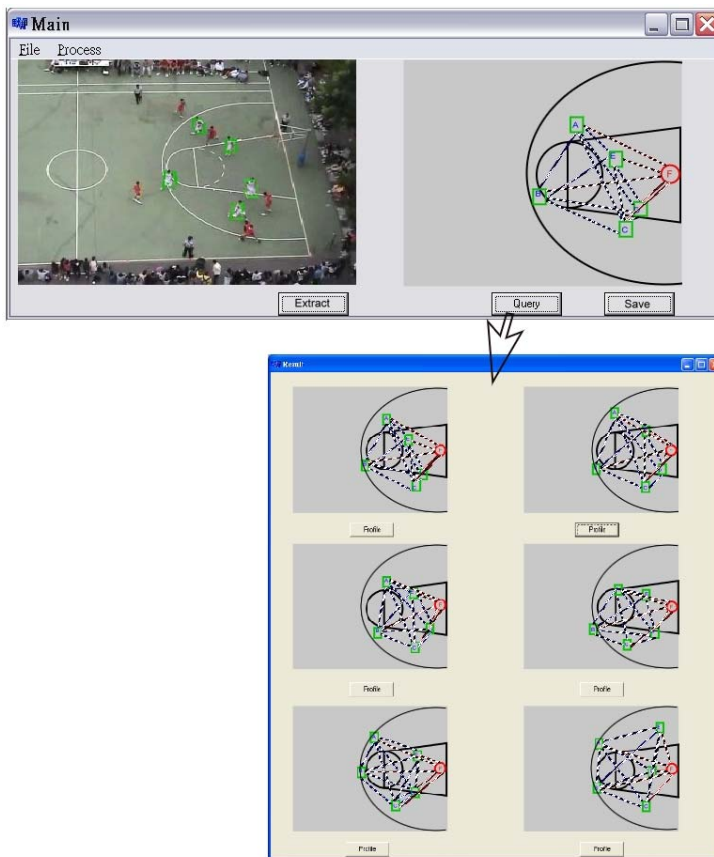


Fig. 6. The GUI and query results

## 5 Conclusion and Future Work

In this paper, we track objects moving in basketball game video sequence and record the locations of the defensive players. After extracting the locations, we used Spatial

Relationships to define the relationships between players for evaluating basketball defensive strategies. The system could retrieve the similar defensive strategies efficiently. It will help coaches and players to learn how they carried out the tactics via continuous frames. The coaches could teach players to learn various defensive strategies within the shortest time through the system and without using a white board on which marker pens or colored magnets are used to demonstrate specific tactics. The next work we will proceed to analyze is offensive tactics. A ball game includes offense and defense both which are crucial to win or lose. In addition, coaches can, by using another program to position correct defensive reactions, evaluate players' understanding towards specific tactics from their chosen defensive positions and moves.

## References

1. Su-Li Chin.: The Strategy of Basketball Games. Bulletin of Tamkang University Physical and Sports. A special Edition, (2001) 99-102.
2. Chun-Yeh Liu, Xing-Liang Luo.: Systematic Teaching on Basketball Games. Bulletin of University Education and Sports. Vol.72. (2004) 4-11
3. Glenn Wilkes.: Basketball, Wm. C. Brown publishers, Dubuque. (1990)
4. Teknomo, K., Takeyama, Y., Inamura, H.: Frame-based tracing of multiple objects. On proceedings of 2001 IEEE Workshop on Multi-Object Tracking, (8 July 2001) 11 – 18
5. Tiejhan Lv, Ozer, B., Wolf, W.: A real-time background subtraction method with camera motion compensation. IEEE International Conference on Multimedia and Expo, ICME '04, Volume 1, (27-30 June 2004) 331 - 334
6. Yang Ran, Qinfen Zheng: Mutiple Moving People Detection from Binocular Sequences. On proceedings of International Conference on Multimedia and Expo, Volume: 2 , (6-9 July 2003) II - 297-300
7. Hwann-Tzong Chen, Horng-Horng Lin, Luh Liu.: Multi-object tracking using dynamical graph matching. CVPR 2001, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Volume 2, (2001) II-210 - II-217
8. James F. Allen.: Maintaining Knowledge about Temporal Intervals. Communications of the ACM, Volume 26, No. 11, (1983)

# Intrinsically Motivated Intelligent Rooms

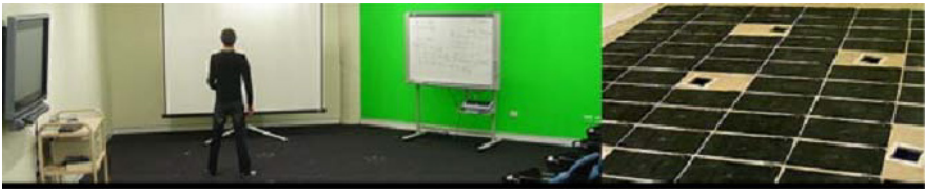
Owen Macindoe and Mary Lou Maher

School of Information Technologies,  
Faculty of Architecture, University of Sydney, Australia  
macindoe@usyd.edu.au, mary@it.usyd.edu.au

**Abstract.** Intelligent rooms are responsive environments in which human activities are monitored and responses are generated to facilitate these activities. Research and development on intelligent rooms currently focuses on the integration of multiple sensor devices with pre-programmed responses to specific triggers. Developments in intelligent agents towards intrinsically motivated learning agents can be integrated with the concept of an intelligent room. The resulting model focuses developments in intelligent rooms on a characteristic reasoning process that uses motivation to guide action and learning. Using a motivated learning agent model as the basis for an intelligent room opens up the possibility of intelligent environments being able to adapt both to people's changing usage patterns and to the addition of new capabilities, via the addition of new sensors and effectors, with relatively little need for reconfiguration by humans.

## 1 Introduction

Developing intelligent rooms, such as The Sentient in the Key Centre for Design Computing and Cognition at Sydney University pictured in Fig 1, has been dominated by the development of configurations of sensors, effectors, and software architectures that specify protocols for interpreting and responding to sensor data.



**Fig. 1.** The KCDCC's intelligent environment, The Sentient

In their seminal papers on IE design, Brookes [1] and Coen [2] argued that a key design goal for developing IEs is to enable them to adapt to, and be useful for, everyday activities. The ability of IEs to adapt their behaviours autonomously to changes in activity patterns is still an open research area. Configuring new sensor and effector systems to allow their IEs to produce useful behaviours is time consuming and labour

intensive. A motivated learning agent, for example, as introduced by Singh et al [3], is an agent that is self-motivated to learn. Self-motivated learning as a basis for an intelligent room creates an IE that is driven to adapt to new sensors and effectors and changing usage patterns. In this paper we present a model for an intrinsically motivated intelligent room that can adapt its learned behaviours from patterns of usage derived from its sensor data.

## 2 Intelligent Environments

An IE is a physical space for living or working that is agent controlled and can bring computational power embedded within it to bear in a manner that helps users of the environment perform their daily tasks. The term Intelligent Environment has not been universally adopted and IEs also go under other names such as Jeng's [4] Ubiquitous Smart Spaces. An IE would necessarily need to be able to sense what is happening inside of it and respond to it with effectors - whether lights, projectors, or doors - in order to exhibit intelligent behaviour and help users.

IE research could be regarded as a sub-field of ubiquitous computing since a major aim of ubiquitous computing is to seamlessly integrate computers into everyday living. IEs have several specific design requirements. Brooks [1] and Coen [2] have argued that IEs should adapt to, and be useful for, ordinary everyday activities; they should assist the user, rather than requiring the user to attend to them; they should have a high degree of interactivity; and they should be able to understand the context in which people are trying to use them and behave appropriately. An IE is essentially, as Kulkarni [5] suggests, an immobile robot, but its design requirements differ from those of normal robots, in that it ought to be oriented towards maintaining its internal space rather than exploring or manipulating an external environment.

MIT's intelligent room prototype e21, shown in Fig 2, facilitates activities via a system called ReBa, described by Hassens et. al. [7] which is the context handling component of the room. ReBa observes a user's actions via the reports of other agents connected to sensors in the room's multi-agent-society and uses them to build a higher level representation of the user's activity. Each activity, such as watching a movie or giving a presentation, has an associated software agent, called a behaviour agent which responds to a user action and performs a reaction, such as turning on the lights when a user enters the room. Behaviours can then layer on top of one another based on the order of user actions, acknowledging differences in context such as showing a presentation in a lecture setting versus a showing one in an informal meeting. Although ReBa can infer context in this way, it cannot adapt to new ways of using the room. In order for an entirely new context to be created, ReBa's behaviour agents must be pre-programmed to recognize the actions of the user and take an appropriate action. It does not self-adapt to new usage patterns. Furthermore, when new sensors are added to the room, the existing rules must be modified manually if they are to take advantage of the new sensor data. Our model, by contrast, uses intrinsic motivation to learn behaviours rather than having the behaviours implemented as part of the agent.

Other researchers have taken approaches to designing environments that are not explicitly agent-based. Both the University of Illinois' Gaia [8] and Stanford University's Interactive Workspace Project [9] have taken a more OS-based approach, de-

veloping Active Spaces and Interactive Workspaces respectively, which focus on the role of the room as a platform for running applications and de-emphasizing the role of the room as a pro-active facilitator. The specification of an action in these systems is triggered by the user and the behaviour is programmed by an applications developer. Gaia's context service provides the tools for applications developers to create agent-based facilitating applications, and the overall model is reactive rather than adaptive. Georgia Tech's Aware Home Research Initiative plans on incorporating an infrastructure for developing context-aware applications [10], but so far no systems exist which allow IEs to self-adapt to new usage patterns. We believe that a motivated agent-based approach allows for this kind of adaptation.



**Fig. 2.** MIT's Intelligent Room Prototype e21, from [6]

### 3 Motivated Learning Agents

In AI literature an agent is anything that can be viewed as perceiving its environment through sensors and then acting within its environment using effectors on the basis of this sensor input. Agent models have a lot in common with IEs: both are described as having sensors for monitoring their environment and effectors for making changes to the environment. A variety of agent models have been developed over time with differing ways of mapping sensor input to effector output, from simple rule-based reactive agents through to complex cognitive agents that try to maintain, and reason about, an internal model of the world. The question then is, what kind of agent model would be a suitable basis for an IE?

An IE needs to be driven to assist users, adapt to changes in its configuration, adapt to changing uses of the IE, and understand context. Drives of this kind have been modelled by the concept of motivation in agent research, leading to several different varieties of motivated agent models. Norman and Long [11, 12, 13] developed a motivated agent model where motivation was modeled by the temporal urgency of tasks to be completed in order for a motivated agent-controlled warehouse to fill orders. Part of the model is shown in Fig 3, which illustrates how the motivation component directed the reasoning process to create new goals for the agent.

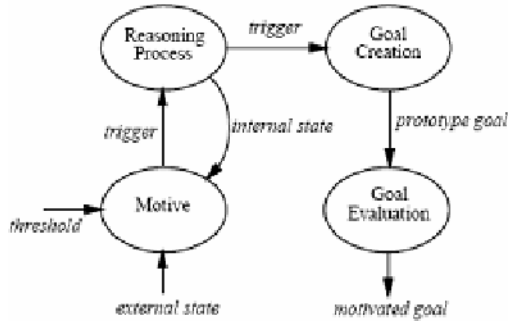


Fig. 3. Norman and Long's motivation model, from [11]

Beaudoin and Sloman [14] developed a simulation of a robot nursery in which a robot nursemaid implementing a motivated agent model was shown to effectively prioritise tasks using a sophisticated model of motivation that included logical propositions, temporal urgency, and levels of insistence. In their design of an agent-controlled water filtration plant, Aylett et al. [15] explicitly extended the role of motivation in their agent model to planning, which showed promise despite the relatively simplistic motivation model used. Kasmarik et al [16] experimented with a domain independent model of motivation based on a novelty detector and used it as a trigger for reinforcement learning in different domain applications.

The requirement for adaptation in an IE can be satisfied with a model of learning new behaviours through the interpretation of sensor data. Rather than specifying a specific set of competencies or goals with an external reward, we look for computational models of novelty and curiosity that allows the agent to respond to unexpected changes in the kinds of activities in the room. Saunders and Gero [17] modeled curiosity computationally as a process that internally generates reinforcement signals that reward the discovery of novelty. They then modeled novelty as the property of being similar enough to other entities of the same class so as to be recognisable as part of that class, but different enough from the norm of that class' form to be unusual. Computationally, novelty was modeled using a self-organizing map that categorized entities presented to the curious agent. The further from the centroid of a class that the new entity's properties lay in the map, the more novel it was considered, but if it were more than a certain threshold away from the centroid the degree of novelty fell off following a Wundt curve, shown in Fig 4, representing dissatisfaction with an entity's "strangeness".

Saunders and Gero demonstrated the utility of this model by using it to simulate the formation of cliques in artistic communities [18], to explore the design space of a simple architectural problem [19], and to provide a richer social force model of human crowds in museums [17]. This model of curiosity as a motivation could be extended for an IE by following the discovery of novelty with learning.

Schmidhuber [18] and Singh et al [3] have developed agent prototypes that are motivated by their own models of curiosity. Schmidhuber developed an agent with a co-evolutionary learning strategy using a highly idiosyncratic model of curiosity that showed promising empirical results in performing exploration when compared with other learning strategies. Singh et. al. developed a model of an intrinsically motivated



reinforcement-learning agent. Inspired by a psychological definition of intrinsic motivation, which is being motivated to do something because it is inherently enjoyable, they developed a learning algorithm in which the learner is rewarded internally for discovering new properties of its domain. They also gave the agent the capacity to build incrementally upon the list of actions that it discovered it could undertake in its domain and allowed them to be chained together into more complicated actions. A comparison between their prototype and a regular reinforcement learning agent showed that it was significantly faster at learning new behaviours. The most interesting feature of the prototype that they built was that the agent was able to learn new behaviours relatively quickly with no human intervention at all. The successes of Schmidhuber and Singh et al.'s motivated learning agents suggest that a motivated learning agent model could be a viable solution to providing the adaptation required for an IE.

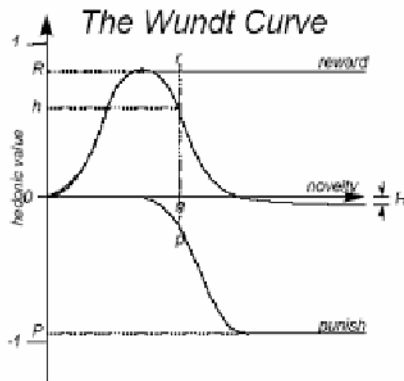


Fig. 4. The Wundt curve. Motivation rises and then falls off as novelty increases. From [16].

### 4 Intrinsically Motivated Intelligent Room

Combining the ideas in IEs with motivated learning agents leads to a model for an intrinsically motivated intelligent room. Motivation can play a valuable role in the agent model for an intelligent room generally, not just in learning, because it provides a model for the pro-active characteristics that are desirable in IEs. We present a motivated agent model for an intelligent room that is motivated by novelty to learn and by competency to act, as illustrated in Fig 5. The model assumes two significant entities: the world and the agent. The world is described at any point in time by the data that can be sensed in the intelligent room. The agent has sensors to sense the state of the world, effectors to change certain aspects of the state of the room, a memory of world states and events, and a reasoning process that includes motivation, action, and learning.

**The World State.** The motivated learning agent exists within a specific world. The state of the world is the basis for agent’s interaction with the world; therefore it becomes the basis for configuring sensors and effectors and adapting to new behaviour patterns. While models and systems for sensors and effectors can be complex hardware and software architectures, we use a simple model here in order to focus on the

agent's reasoning process. The world state at time  $t$ ,  $W(t)$ , is characterised as a partitioned tuple of sensor inputs, which are in turn represented as attribute-value pairs such as `PRESSURE_PAD=ON`. One side of the partition represents inputs from sensors without associated effectors, such as a pressure pad in the floor. The other side of the partition represents inputs from sensors that do have associated effectors, such as a sensor attached to a light switch which can be both activated manually by a human operator and automatically by the room itself. A world state representation  $W(t)$  will therefore take the form:

```
W(t) ::= <senseData>
<senseData> ::= "(" <senseOnly> "|" <senseEffect> ")"
```

And an example of such a representation is:

```
W(0) = (PRESSURE_PAD=ON | LIGHT_DIMMER_INTENSITY=0.5,
DESK_LAMP=ON)
```

This distinction is relevant because the intention is for the motivated agent to learn behavioural rules that include changes in the effectable sensor data part of its sufficient conditions. For instance a rule such as the following would represent a behaviour that the IE would not have the capacity to enact since it does not have the effectors necessary to achieve it:

```
IF SENSE = (PRESSURE_PAD=ON) THEN EFFECT =
(PRESSURE_PAD_4=ON)
```

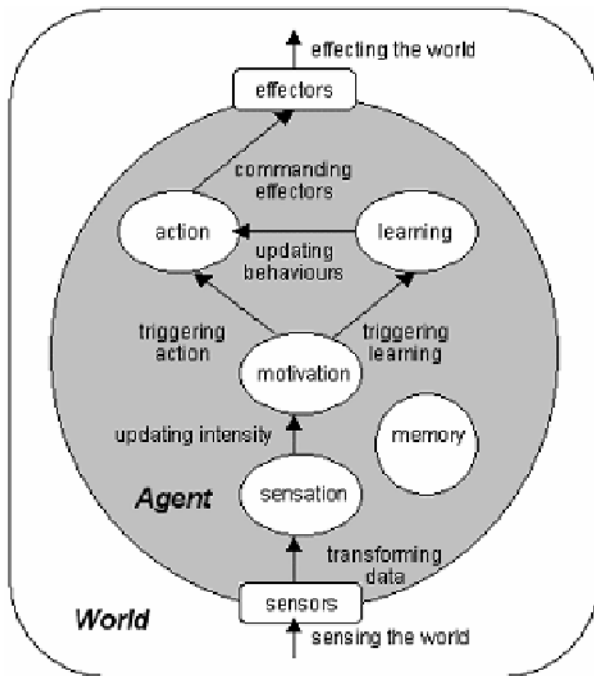


Fig. 5. The intrinsically motivated intelligent room model

**Sensation.** In the sensation process, sensor input from the world is converted into a form suitable for performing reasoning and learning. The new world state  $W(t)$  is stored in the set  $S$  of recent world states sensed by the agent. The sensation component also records events or **changes** in the world state. An event is represented as  $\Delta(t)$ , the changes in sensor inputs between  $W(t)$  and  $W(t-1)$ .  $\Delta(t)$  takes the same form as  $W(t)$ , a partitioned tuple, but the values of the tuple represent the change in value between  $W(t)$  and  $W(t-1)$  with numeric values being calculated as normalized differences and nominal elements being 0 if no change occurred and 1 if one did occur. For example:

$$W(0) = (\text{PRESSURE\_PAD}=\text{ON} \mid \text{LIGHT\_DIMMER\_INTENSITY}=0.5, \\ \text{DESK\_LAMP}=\text{ON})$$

$$W(1) = (\text{PRESSURE\_PAD}=\text{OFF} \mid \text{LIGHT\_DIMMER\_INTENSITY}=0.8, \\ \text{DESK\_LAMP}=\text{ON})$$

$$\Delta(1) = (\text{PRESSURE\_PAD}=1 \mid \text{LIGHT\_DIMMER\_INTENSITY}=0.3, \\ \text{DESK\_LAMP}=0)$$

The sensation component recognises new sensors as an event in the  $\Delta(t)$  tuple.  $\Delta(t)$  is converted to a set of event labels,  $\{e_1 \dots e_n\}$ , that occurred at time  $t$ . The event labels are the basis for motivation, learning, and acting.

**Motivation.** The intelligent room is motivated to learn when it recognizes a novel event. In the beginning, everything is novel and the agent is motivated to learn rather than act. As the agent builds a set of behaviours, it is motivated to act when it recognizes an event that triggers a known behaviour and to learn when it recognizes a novel event. Our current novelty detector is based on a model of “interesting” developed and implemented by Kasmarik et al [16] for a motivated agent model. In this model, an event is interesting if it is rare in the agent’s cumulative experience of the world. This suits our need for identifying a novel event. Events are divided into groups using unsupervised clustering of event frequencies. Each group is defined to be novel or not novel based on their frequencies of occurrence. The novel events are then further clustered into groups of increasing rarity so that the agent can be motivated to learn about more common or ‘easier’ events that are more likely to have sufficient patterns in the agent’s memory.

Clustering is performed by first sorting events in order of ascending frequency where frequency is calculated as the number of times the event has occurred divided by the size of the agent’s lifetime. This produces an ordering  $(e_1, f_1), (e_2, f_2) \dots (e_n, f_n)$  with differences  $d_1, d_2 \dots d_n$  where  $d_k = f_k - f_{k-1}$ . K-means clustering with  $k=2$  and initial centroids 0 and  $d_{max}$  where  $d_{max} = \max d_j$  produces two groups  $g_1$  and  $g_2$  with average distances to centroids  $a_1$  and  $a_2$ .  $g_i$  has the minimum average  $a_i$ , then events can be clustered as follows: Place  $f_1$  in a new cluster. For  $f_2, f_3 \dots f_n$ , place  $f_k$  in the same cluster as  $f_{k-1}$  if  $d_k \in g_i$  or in a new cluster otherwise. We say that an event  $e_i$  is novel if its frequency  $f_i$  falls in the same cluster as  $f_1$ .

**Learning.** Learning must rely on finding patterns in previously experienced world states since it is inappropriate for an intelligent room to experiment with changes in the state of the room. The aim of the learning component of the agent model is to infer a set  $R$  of behavioural rules from the set of stored world data  $S$  and then store  $R$  in memory for the action component to utilize. Such behavioural rules will be of the form:

Rule ::= IF SENSE = <window> THEN EFFECT = <action>

Where <window> is a tuple of event label and time pairs satisfying a constraint on  $t$  and <action> is a tuple of event labels relating to effectors. Such rules are formed by considering the changes in world state within a given time window and constructing rules to enact equivalent changes when sufficient support and confidence levels exist for such a rule to be derived. Data mining techniques such as MINEPI mining can find these rules from the memory of event labels.

**Action.** The action component of the agent model maps the most recently sensed world state  $W(t)$  and previous world states within a given time window to a rule from the set of behavioural rules  $R$  to be executed by the IE's effectors. It then sends the appropriate commands to the IE's effectors to enact the changes in the world dictated by the rule selected.

**Memory.** The sensation, motivation, learning, and action components all require information about earlier states of the world, and all except action update that information. The memory component of the agent comprises a representation of previous worlds states, deltas, events, and behavioural rules.

## 5 Conclusions

A model for an intelligent room based on an intrinsically motivated learning agent moves us closer to an adaptable intelligent environment. Our initial tests with this model include sensor data that identifies different behaviours associated with the location of people in the room (the pressure pads) and the state of the electric devices in the room (lights, projectors, applications being projected). Given this kind of data, behaviours can be learned that are based on patterns of use, rather than on the identities of the individuals in the room. We are currently simulating the sensor data based on activity scenarios to test the appropriateness of our novelty detector and rule mining algorithms. The validation of this model is a test of its adaptability, that is, can the room change its behaviour when new sensor or effector data are introduced.

## References

1. Brooks, R.A., Coen, M., Dang, D., DeBonet, J., Kramer, J., Lozano-Perez, T., Mellor, J., Pook, P., Stauffer, C., Stein, L., Torrance, M., Wessler, M.: The Intelligent Room Project. In: Proceedings of the Second International Cognitive Technology Conference (CT'97). Aizu, Japan (1997) 271-279
2. Coen, M.H.: Design Principles for Intelligent Environments. In: Proceedings of the Fifteenth National / Tenth Conference on Artificial Intelligence / Innovative Applications of Artificial Intelligence. Madison, Wisconsin, United States (1998) 547-554
3. Singh, S., Barto, A.G., and Chentanez, N.: Intrinsically Motivated Reinforcement Learning. <http://www.eecs.umich.edu/~baveja/Papers/FinalNIPSIMRL.pdf> Accessed 7/4/2004 (2004)
4. Jeng, T.: Designing a Ubiquitous Smart Space of the Future: The Principle of Mapping. In: Gero, J.S. (ed.): Design Computing and Cognition '04. Kluwer Academic Publishers, The Netherlands (2004) 579-592

5. Kulkarni, A.: Design Principles of a Reactive Behavioral System for The Intelligent Room. In: *Bitstream: The MIT Journal of EECS Student Research*, April 2002 Edition. Cambridge, MA (2002) 1-5
6. Kottahachchi, B., Laddaga, R.: Access Controls for Intelligent Environments. In: *Proceedings of ISDA '04: 4th Annual International Conference on Intelligent Systems Design and Applications*. Budapest, Hungary (2004)
7. Hanssens, N., Kulkarni, A., Tuchinda, R., Horton, T.: Building Agent-Based Intelligent Workspaces. In: *Proceedings of the 3rd International Conference on Internet Computing*, (2002) 675-681
8. Román, M., Hess, C.K., Cerqueira, R., Ranganathan, A., Campbell, R.H., Nahrstedt, K.N.: Gaia: A Middleware Infrastructure to Enable Active Spaces. In: *IEEE Pervasive Computing Oct-Dec (2002)* 74-83
9. Johanson, B., Fox, A., Winograd, T.: The Interactive Workspaces Project: Experiences With Ubiquitous Computing Rooms. In: *IEEE Pervasive Computing Magazine*, vol. 1, no. 2, April-June (2002) 67-74
10. Kidd, C., Orr, R., Abowd, G., Atkeson, C., Essa, I., MacIntyre, B., Mynatt, E., Starner, T., Newstetter, W.: The Aware Home: A Living Laboratory for Ubiquitous Computing Research. In: *Proceedings of the Second International Workshop on Cooperative Buildings*. Pittsburgh, PA (1999)
11. Norman, T.J., Long, D.P.: Goal Creation in Motivated Agents. In: Wooldridge, M.J., Jennings, N.R. (eds.): *Intelligent Agents: Proceedings of the First International Workshop on Agent Theories, Architectures and Languages*, Volume 890 of *Lecture Notes in Artificial Intelligence*, Springer-Verlag (1995a) 277-290
12. Norman, T.J. and Long, D.P.: Alarms: Heuristics for the Control of Reasoning Attention. In: Moore, J. and Lehman, J. (eds.): *Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society*, Pittsburgh, PA (1995b) 494-499
13. Norman, T.J. and Long, D.P.: Alarms: An Implementation of Motivated Agency. In: Wooldridge, M.J., Müller, J.P., Tambe, M. (eds.): *Intelligent Agents II: Proceedings of the Second International Workshop on Agent Theories, Architectures, and Languages*, Volume 1037 of *Lecture Notes in Artificial Intelligence*, Springer-Verlag, (1996) 219-234
14. Beaudoin, L.P., Sloman, A.: A Study of Motive Processing and Attention. In: Sloman, A., Hogg, D., Humphreys, G., Partridge, D., Ramsay, A. (eds.): *Prospects for Artificial Intelligence*. IOS Press, Amsterdam (1993) 229-238
15. Aylett, R.S., Coddington, A.M., Petley, G.J.: Agent-Based Continuous Planning. <http://planning.cis.strath.ac.uk/plansig/pastSIGs/open-univ-19/aylett.pdf> Accessed 7/4/2005 (2000)
16. Kasmarik, K., Uther, W. and Maher, M.-L.: Motivated Agents, In: *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*. Edinburgh, Scotland (2005) 1505-1506
17. Saunders, R., Gero, J.S.: Designing for Interest and Novelty: Motivating Design Agents. In: de Vries, B., van Leeuwen, J., Achten, H. (eds.): *CAADFutures 2001*. Kluwer, Dordrecht (2001) 725-738
18. Saunders, R., Gero, J.S.: Curious Agents and Situated Design Evaluations. In: Gero, J.S., Brazier, F. (eds.): *Agents in Design 2002*. Key Centre of Design Computing and Cognition, University of Sydney, Australia (2002) 133-149
19. Saunders, R., Gero, J.S.: The Digital Clockwork Muse: A Computational Model of Aesthetic Evolution. In: Wiggins, G. (ed.): *Proceedings of the AISB'01 Symposium on AI and Creativity in Arts and Science*. University of York, York, UK (2001) 12-21
20. Schmidhuber, J.: What's interesting? Technical Report IDSIA-35-97. (1997) Lugano, Switzerland

# Multivariate Stream Data Reduction in Sensor Network Applications

Sungbo Seo<sup>1,\*</sup>, Jaewoo Kang<sup>2</sup>, and Keun Ho Ryu<sup>1</sup>

<sup>1</sup> Dept. of Computer Science, Chungbuk National University, Chungbuk, Korea  
{sbseo, khryu}@dblabb.cbu.ac.kr

<sup>2</sup> Dept. of Computer Science, North Carolina State University, Raleigh, NC, USA  
{kang}@csc.ncsu.edu

**Abstract.** We evaluated several multivariate stream data reduction techniques that can be used in sensor network applications. The evaluated techniques include Wavelet-based methods, sampling, hierarchical clustering, and singular value decomposition (SVD). We tested the reduction methods over the range of different parameters including data reduction rate, data types, number of dimensions and data window size of the input stream. Both real and synthetic time series data were used for the evaluation. The results of experiments suggested that the reduction techniques should be evaluated in the context of applications, as different applications generate different types of data and that has a substantial impact on the performance of different reduction methods. The findings reported in this paper can serve as a useful guideline for sensor network design and construction.

## 1 Introduction

A typical wireless sensor network (WSN) consists of small battery-powered wireless devices and sensors. Conserving battery power on such devices is crucial to improve the life span of a WSN. Among many operations that a sensor node performs, transmitting data among sensor nodes typically consumes the most energy. Many data reduction techniques have been proposed to address this problem [1, 2, 3]. However, different sensor networks have different data requirements depending on the types of applications they run and characteristics of data generated by different applications can be also different. Thus, such data reduction techniques need to be evaluated in the context of applications and the types of data they generate. In this paper, we attempt to identify such application specific requirements, and to propose different data reduction techniques for different types of application scenarios.

Three broad areas of sensor network applications are *environmental monitoring*, *object tracking*, and *object guarding* [4, 5, 6, 9]. First, examples of environmental monitoring are flood detection, home application and habitat monitoring. Long-term data analysis over low frequency data is usually used in this type of applications. Second, examples of object tracking include vehicle tracking, military applications and SCM (Supply Chain Management). These applications typically generate high

---

\* Work performed while the author visited North Carolina State University.

frequency multivariate data. Finally, examples of object guarding are emergency medical care, intrusion detection and earthquake risk assessment. These applications require real-time monitoring of outliers and detection of abnormality in the data. As we see here, different applications need different models for data acquisition, transmission, and storage. These need to be considered together with physical constraints such as limited bandwidth and power, and unreliable network, when the data reduction techniques are evaluated.

A typical sensor network example is shown in Fig. 1. A sensor node has one or more sensors. A node periodically collects data from its own sensors as well as data transmitted from other children sensor nodes. Thus, data collected by a sensor node naturally forms a multivariate time series. Previous researches on data acquisition and transmission have suggested data reduction techniques suitable for single or relatively small numbers of attributes [2, 7]. However, these techniques may not be suitable for applications such as object tracking and guarding as they typically generate multivariate data with large numbers of attributes. This problem is even more exacerbated in sink nodes (see Fig. 1) where data generated by all sensor nodes in the network is collected and aggregated.

In this work, we studied efficient, multivariate approximate data transmission techniques as follows. First, we defined the hierarchical/distributed sensor network architecture and data model. Second, we classified application areas in wireless sensor networks, and then briefly introduced the multivariate data reduction techniques, such as Wavelet, HCL (Hierarchical Clustering), Sampling and SVD (Singular Value Decomposition). Finally, we experimented with data reduction methods with respect to relative error and reduction ratio.

The rest of the paper is organized as follows. Section 2 presents related work. Section 3 defines a hierarchical/distributed sensor network architecture and data model. In section 4 we suggest a simulation model and introduce some multivariate data reduction techniques. Section 5 reports the result of our experiments. Section 6 presents concluding remarks.

## 2 Related Work

Many previous works [1, 2, 3] in sensor networks studied data aggregation and approximate data transmission between sink nodes and base stations. Generally, data analysis and reduction techniques in sensor networks include clustering, wavelet, histogram, regression, aggregation, sampling, PCA and SVD. Aggregation is an effective mean to get a synopsis (avg., max., min.), but is rather crude for applications that need detailed historical information [3]. Spectral models such as DWT, DFT and DCT are tuned for time sequence, ideally with a few low-frequency harmonics, but it is ineffective under the multi-dimensional attributes [11, 13]. Sampling has a good performance, but has some problems such as sampling ratio, relational join over arbitrary schemas and set-valued approximate queries [10, 11]. Clustering techniques for stream data is presented in [15] which analyzed the complexity and requirements of one-pass clustering over streaming data.

These previous works focused on solving the problems with intrinsic characteristics and limitations of sensor networks, but these techniques don't take into account the

application specific requirements and different types of nodes with varying capabilities. In this paper, we evaluated the multivariate data reduction methods in the context of different applications. The findings reported in this paper can serve as a useful guideline for sensor network design and construction.

### 3 System Architecture

Hierarchical/Distributed organization is the most widely adopted model in sensor network [4]. Fig. 1 shows its architecture. Each type of nodes has the following characteristics.

- Sensor node gathers periodically the multivariate data collected from sensors or target nodes. Data transmission is done by a multi-hop or cluster-based communication method and not typically done by a point-to-point direct communication.
- Each sensor node has a small processor and main memory, and periodically sends the data to sink node by wireless communication. Sink node collects the data from the nodes and usually contains in-memory DBMS.
- Sink nodes transmit data to a base station through a wireless communication. Aggregated data collected in a base station can be stored in a server node for archiving and for serving historical queries spanning over long period of time.
- Server node and base station use an existing network infrastructure and have a traditional DBMS. Generally, a server node has a multi-dimensional data cube in order to serve aggregate queries efficiently.

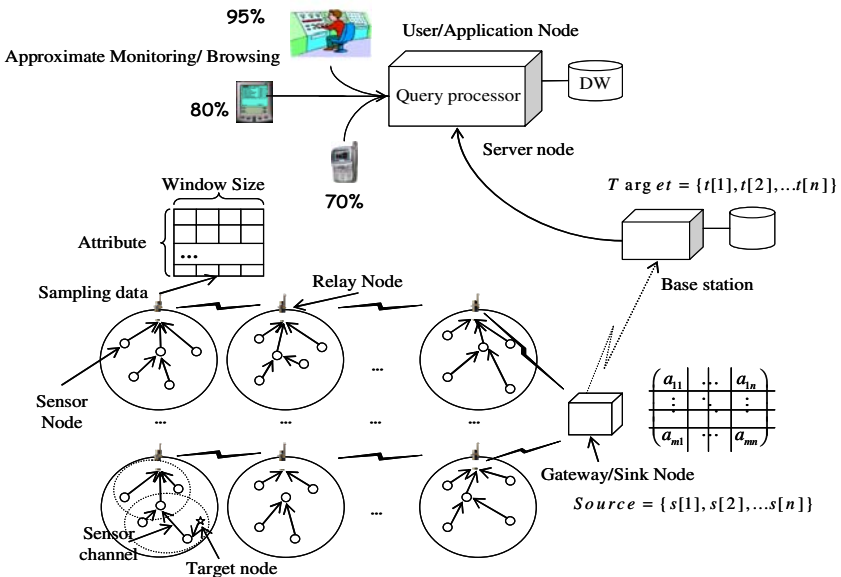


Fig. 1. General architecture and simulation model in wireless sensor network



As shown in Fig. 1, data collected by each sensor node is transmitted to the sink node. The sink node then temporarily stores the data for some time, and periodically sends the data to the base station. Data reduction is typically performed in this transmission because the size of aggregated data can be large, and depending on the applications, often times large, exact original data is out of favor to compact approximate summarization [8].

Communication between the base station and the server node typically use a wired network such as LAN, and hence the data transmission and reduction methods for these nodes should be considered differently. Unlike sensor and sink nodes, these nodes contain a powerful CPU, a large amount of memory, and reliable power sources. Efficient query processing over the large collection of aggregated data should be the more important consideration in these nodes. Similar to the transmission model, the query and data acquisition model also have to be determined according to the application requirements.

## 4 Multivariate Data Reduction Methods in Sensor Networks

We compared the multivariate data reduction methods, such as DWT (Discrete Wavelet Transformation), HCL (Hierarchical Clustering), Sampling, and SVD (Singular Value Decomposition) over different types of data generated from different application scenarios. In what follows, we present brief descriptions of the data reduction techniques and their characteristics.

**DWT:** The DWT is a linear signal processing technique using a hierarchical decomposition function. DWT is closely related to the DFT (Discrete Fourier Transform) and performs well with a low frequency data type. However, its performance degrades if data has several spikes or abnormal jumps [10, 13]. The advantages of DWT are the fast computation and small space complexity. A fast DWT algorithm has a complexity of  $O(n)$  for an input vector of length  $n$  [10]. Some researchers [7, 11] proposed the improved versions of the wavelet method, but it is still inefficient with the presence of multi-dimensional attributes.

**HCL:** Clustering is partitioning the objects into groups or clusters so that objects within a cluster are similar to one another and dissimilar to objects in other clusters [10, 15]. It can be used for data reduction as a group of similar objects in a cluster can be replaced with a single centroid. In order to cluster multivariate data set, in our experiments, we used the hierarchical clustering method using single, average and complete-linkage method. The HCL with multi-dimensional index tree can be used for hierarchical data reduction as well as for the fast approximate answers to queries.

**Sampling:** Sampling can be used as a data reduction technique since it allows a large data set to be represented by a much smaller random sample of the data [10, 11]. An advantage of sampling for data reduction is that the cost of obtaining a sample is proportional to the size of the sample. The complexity of sampling is potentially linear and we can easily control sampling rate according to the error ratio. But it is ineffective for ad-hoc relational joins over arbitrary schema and effectiveness for set-valued approximate queries is unclear [11].

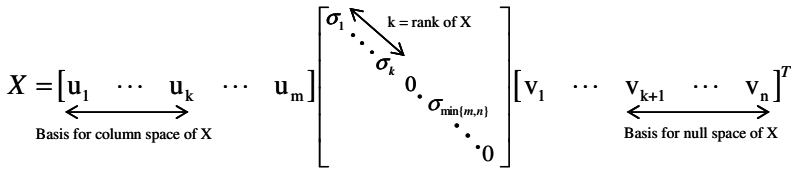
**SVD:** SVD can be used for multivariate data reduction and is defined as follows.

**Definition 1. (SVD):** Given an  $m \times n$  real matrix  $X$ , we can express it as  $X=U\Sigma V^T$  where  $U$  and  $V$  are column-orthonormal and  $\Sigma$  is a diagonal matrix such that

$$U_{m \times m} = UU^T = U^T U = I, V_{n \times n} = VV^T = V^T V = I \tag{1}$$

$$\Sigma_{m \times n} = [\Sigma]_{ij} = 0, i \neq j, [\Sigma]_{ii} = \sigma_i \geq 0, \sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{\min\{m,n\}} \tag{2}$$

Recall that a matrix  $U$  is called column-orthonormal if its columns  $u_i$  are mutually orthogonal unit vectors. So,  $U^T$  is equal to  $U^{-1}$  and  $U \times U^T=I$ , where  $I$  is the identity matrix.  $\Sigma$  is a diagonal matrix with values called singular values  $\{\sigma_i\}$  in its diagonal. The rank  $k$  of  $X$  equals to the number of nonzero singular values of  $X$ . The SVD of  $X=U\Sigma V^T$  can be illustrated as follows.



**Fig. 2.** Column space, rank and null space

As for the space complexity, the original matrix  $X$  contains  $N \times M$  data elements while the SVD representation, after truncating to  $k$  principal components, will need  $N \times k$  data elements for  $U$ ,  $k$  data elements for the Eigen values, and  $k \times M$  data elements for the  $V$  matrix. Thus, the reduced data to the original data ratio,  $s\_ratio$ , is as follows [12, 13].

$$s\_ratio = \frac{N \times k + k + k \times M}{N \times M} \approx \frac{k}{M} \quad (N \gg M \geq k) \tag{3}$$

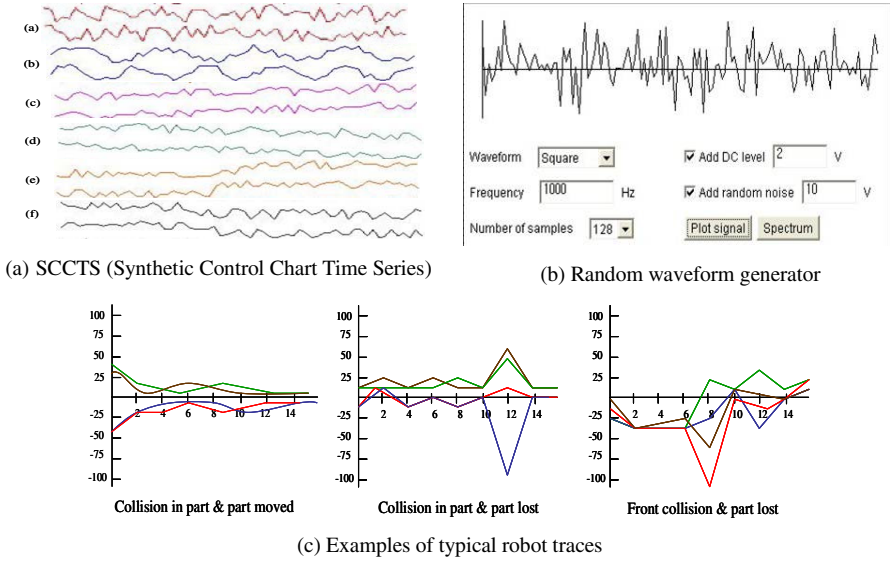
## 5 Experiments and Analysis

### 5.1 Data Sets

Our results are based on experiments over three data sets obtained from [16, 19]. The first data set, SCCTS (Synthetic Control Chart Time Series), contains 600 examples of control charts synthetically generated by the process introduced by Alcock and Manolopoulos in [16].

The SSCTS consists of the six different classes of control charts (Normal (a), Cyclic (b), Increasing trend (c), Decreasing trend (d), Upward shift (e), Downward shift (f)). The second data set include five synthetic data sets generated using the waveform generator. Each data set is created applying different combinations of parameters including waveform (one of sine, cosine, square, and saw-tooth), frequency (in Hz), DC level and random noise [19]. The third data set is the robot traces containing force

and torque measurements on a robot moving an object from one location to another. Each movement is characterized by 15 force/torque samples collected at regular time intervals [14, 16].



**Fig. 3.** Data sets of multivariate time series and sensor data

In order to measure the relative error ( $\sigma$ ) between the original matrix  $A$  and its approximation  $\hat{A}$ , we used the following metric.

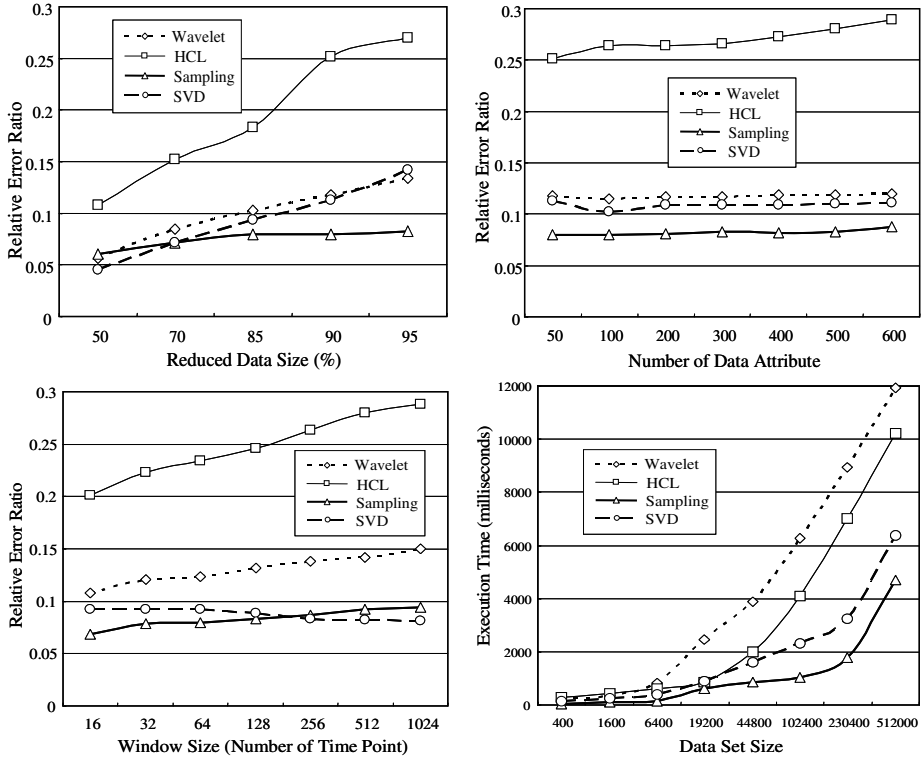
$$\sigma = \frac{\|\hat{A} - A\|_F}{\|A\|_F}, \text{ where } \|A\|_F = \left( \sum_{ij} |a_{ij}|^2 \right)^{1/2} \quad (4)$$

In order to compute the relative errors for the reduction methods, we need to be able to recover the original matrix from its reduced form. The recovered matrix,  $\hat{A}$ , is an approximation of the original matrix,  $A$ , and has the same dimensionality as  $A$ . Computing  $\hat{A}$  is straightforward for all reduction methods by their definition except the sampling method. We interpolated the sample points to approximate the missing values in the time points where the samples were not taken. For the experiments, we used the multivariate data reduction algorithms available from [17, 18] after some modification.

## 5.2 SSCTS Data (Data Size vs. Performance)

Fig. 4 shows the result of experiments where we compared the relative errors of the reduction methods over the range of different parameters. Fig. 4 (top left) compares the relative errors over the range of different data reduction ratios from 50% to 95% (e.g., 95% means the size of data after reduction is just 5% of the original). HCL was

the worst performer while sampling showed the best performance. Fig. 4(top right) compared the reduction methods over the varying numbers of attributes (or dimensions) in the input data. For example, at  $x=50$  (the first data point in the x axis), the algorithms are compared over data with 50 sensor readings in each time point. In this test and the next test (shown on the bottom left), we fixed the reduction ratio to 90%. As the figure shows, all methods are not affected much by dimension size.



**Fig. 4.** Data size vs. Reduction methods performance

Fig. 4(bottom left) shows if the data window size has any influence on the performance of the methods. In each sensor node, data is accumulated for a while before transmitted to the node in the upper layer. The window size determines how much readings will be accumulated for each transmission. For example, if the window size is 10, then sensor readings are accumulated for 10 time points and transmitted as a unit. In this test, SVD showed a stable performance over the increasing window sizes while the others, especially HCL and Wavelet, showed increasing errors for larger windows.

Fig. 4(bottom right) compares the execution time for each method as the data size increases. This result shows that HCL and wavelet are more computationally expensive than others. Overall, sampling was superior to others for six different classes in SSCTS. Wavelet took longer than others and was susceptible to the increase of window size. SVD showed a reliable performance in most of the cases.

### 5.3 Synthetic Data (Data Type vs. Relative Error Ratio)

Fig. 5 compares the performance of the data reduction methods over the different types of data generated from different application areas. The synthetic data set generated from the waveform generator was used. In order to emulate the object tracking and object guarding scenarios, we inserted randomly generated outliers to the data. In this experiment, we fixed the data reduction ratio to 80% while varying the window size and the number of attributes. Fig. 5 (top left) shows the result with low frequency data set such as sine or cosine curves having low harmonic characteristics in the same attribute. All methods performed well in this test except HCL. HCL failed to produce comparable results.

Fig. 5 (top right) shows the result with the high frequency data set. HCL was the worst while sampling was the best. SVD and Wavelet performed reasonably well. Fig. 5 (bottom left) shows the result with the mixed input data with the ratio of high frequency to low frequency being 3:2. Fig. 5 (bottom right) shows the result with the data set containing outliers and abnormal patterns. SVD performed well while HCL and wavelet did not.

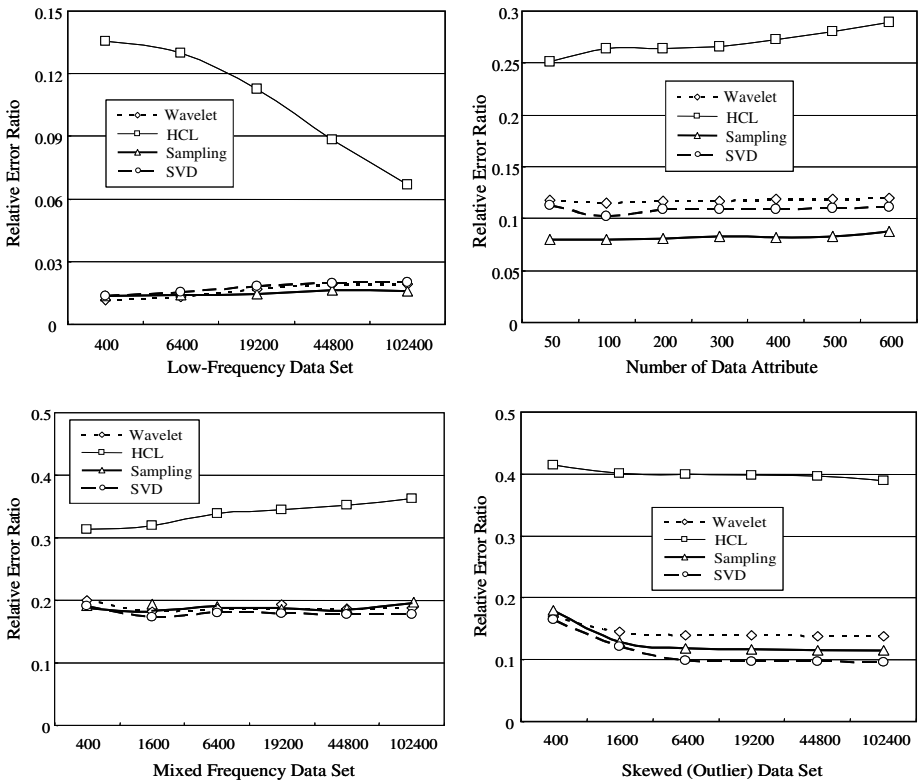


Fig. 5. Data types vs. Relative error ratio

### 5.4 Robot Trace Data (SVD vs. Adaptive Reduction)

Fig. 6 shows the results of experiments performed with the robot trace data (obtained from [14, 16]). In Fig. 6 (left), we compared the four methods over five different types of trace data including Normal, Collision, Obstruction, Lost, and Move as described in [16]. The reduction rate is fixed to 80% in this test. Overall, SVD showed more stable performance than others throughout the test. Fig. 6 (right) compares the SVD method (the best performer in the previous test) with the adaptive reduction method where we apply the reduction method adaptively for each window. The data set used in this test also has five different types of traces, represented as LP1 to LP5 as described in [16].

In this adaptive method, data in each window is first examined and the best reduction method for the given window is determined and applied. In order to implement this approach correctly, we need a classifier that predicts the labels for each window characterizing the properties of data in the window. Although it is an interesting and important area of research, exploring multivariate classifiers is out of scope of this paper. In our implementation of the adaptive approach, we simply assumed the correct labels for each window are given. As the result suggests, given an accurate classifier, we can achieve a significant improvement on the reduction performance over the static methods. We plan to investigate this adaptive reduction framework in our future work.

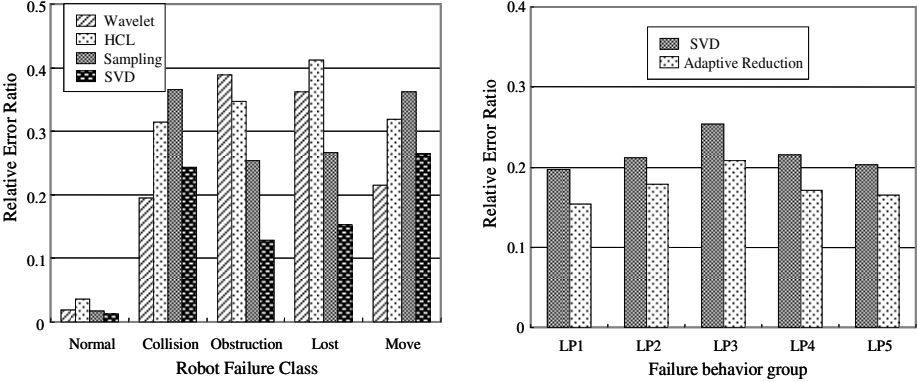


Fig. 6. Relative error ratio of robot failure behavior

## 6 Conclusion

We compared multivariate data reduction techniques that can be used in various sensor network applications, including wavelet, HCL, sampling and SVD methods, over both the real and synthetic time series data. We showed the relative performance of different methods vary over the data sets with different data characteristics. The findings reported in this paper can serve as a useful guideline for sensor network design and construction.

**Acknowledgement.** This work was partially supported by the RRC Program of MOCIE and ITEP, also ETRI (Telematics & USN Research Division) in Korea.

## References

1. J. M. Hellerstein, W. Hong, and S. R. Madden.: The Sensor Spectrum: Technology, Trends, and Requirements. In SIGMOD Record (2003) 22-27
2. A. Deligiannakis, Y. Kotidis and N. Roussopoulos.: Compressing Historical Information in Sensor Networks. In SIGMOD (2004) 527-538
3. A. Deligiannakis, Y. Kotidis, and N. Roussopoulos.: Hierarchical in-Network Data Aggregation with Quality Guarantees. In EDBT (2004) 658-675
4. M. J. Franklin and S. R. Jeffery et al.: Design Considerations for High Fan-In Systems: The HiFi Approach. In CIDR (2005) 290-304
5. A. Mainwaring and J. Polastre et al.: Wireless Sensor Networks for habitat monitoring. In WSNA (2002) 88-97
6. B. X. and O. Wolfson.: Time-Series Prediction with Applications to Traffic and Moving Objects Databases. In MobiDE (2003) 56-60
7. S. Guha, C. Kim and K. Shim.: XWAVE: Approximate Extended Wavelets for Stream Data. In VLDB (2004) 288-299
8. A. Deshpande and C. Guestrin et al.: Model-Driven Data Acquisition in Sensor Networks. In VLDB (2004) 588-599
9. R. C. Oliver and K. Smettem et al.: Field Testing a Wireless Sensor Network for Reactive Environmental Monitoring. In ISSNIP (2004) 7-12
10. 10 J. Han and M. Kamber.: Data Mining Concepts and Techniques. Morgan Kaufmann Publishers (2000)
11. M. Garofalakis, and P. B. Gibbons.: Approximate Query Processing: Taming the Terabytes! In VLDB Tutorial (2001)
12. G Strang, Introduction to Linear Algebra, 3<sup>rd</sup> Edition, Wellesley-Cambridge Press (1998)
13. F. Korn, H. V. Jagadish and C. Faloutsos.: Efficient Supporting Ad Hoc Queries in Large Datasets of Time Sequences. In ACM-SIGMOD (1997) 289-300
14. L. M. Camarinha-Matos, L. S. Lopes, and J. Barata.: Assembly Execution Supervision with Learning Capabilities. In ICRA (1994) 272-279
15. S. Guha and N. Mishra et al.: Clustering Data Streams. In FOCS (2000) 359-366
16. S. Hettich, and S. D. Bay.: The UCI KDD Archive (Synthetic Control Chart Time Series, Robot Execution Failures) [<http://kdd.ics.uci.edu>]. Irvine, CA: University of California, Department of Information and Computer Science (1999)
17. JAMA: A Java Matrix Package: <http://math.nist.gov>
18. Multivariate Data Analysis Software: <http://astro.u-strasbg.fr/~fmurtagh/mda-sw/>
19. FFT Spectrum Analyzer: <http://www.dsptutor.freeuk.com/analyser/SA102.html>

# Implementing a Graph Neuron Array for Pattern Recognition Within Unstructured Wireless Sensor Networks

M. Baqer, A.I. Khan, and Z.A. Baig

Monash University, Australia

{mohamed.baqer, asad.khan, zubair.baig}@infotech.monash.edu.au

**Abstract.** Graph Neuron (GN) is a network-centric algorithm which envisages a stable and structured network of tiny devices as the platform for parallel distributed pattern recognition. However, the unstructured and often dynamic topology of a wireless sensor network (WSN) does not allow deployment of such applications. In this paper, using GN as a test-bed application, we show that a simple virtual topology overlay would enable distributed applications requiring stable structured networks to be deployed over dynamic unstructured networks without any alteration.

## 1 Introduction

WSNs are deployed in critical environments for event sensing and reporting purposes. Due to the computation constraints of sensors, most contemporary WSNs are designed to react to immediate real-time events by relying on a high-performance base station or a server for centralised event processing. Such WSNs have been deployed in areas such as Health Monitoring, Traffic Control, and Industrial Sensing. More recent applications are in Infrastructure Security and other detection/tracking areas [1]. The Graph Neuron (GN) proposed in [2][3], as a real-time parallel pattern recognition algorithm for tiny devices uses a simple distributed algorithm to intelligently recognise patterns in the monitored environment. We assert that for real-time systems, particularly those which make use of in-network processing, the underlying network must have a structured topology with established mechanisms for ensuring bounded time delays for all of its operations. Chord [4] as distributed lookup scheme can be conjectured to provide a structure to a WSN for seamlessly handling the dynamic aspects of the WSN and for providing the necessary framework for supporting distributed applications such as the GN. In this paper we would investigate a Chord based scheme for establishing a self-organising and self-adapting structure for the WSN. Additionally, we would use the Chord overlay to meet the application's scalability requirements.

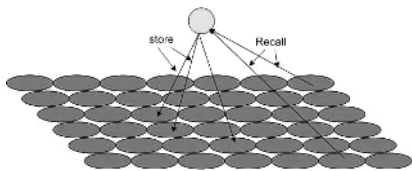
## 2 Background

Graph Neuron (GN) is a pattern recognition algorithm which can form an associative memory network by interconnecting tiny devices in a graph-like structure

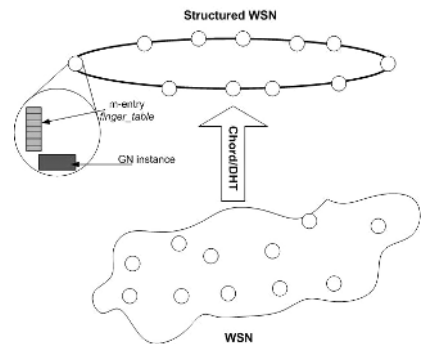


called the GN array; shown in Figure 1. The information presented to each of the tiny devices (GN nodes) is in the form of a value, position pair. Each of these pairs represents a data point in the reference pattern space. The GN array hence forth converts the spatial/temporal patterns into a graph-like representation and then compares the edges of the graphs for memorisation and recall operations. The advantage of having a graph-like representation is that it provides a mechanism for placing the spatial/temporal information in a context. Hence not only can we compare the individual data points but we may also compare the order in which these occur. The GN algorithm utilises the real parallelism present within the network to speed up the comparisons. The number of comparisons required for matching a stored pattern with an incoming sequence for an  $n$  vertex graph could be of the order of  $O(2^{n/3})$ [2][5], however the GN circumvents this very high computational cost through fine-grained parallelism [5].

The data representation for a GN may be summarised as follows: input pairs  $p(\text{value,position})$  are mapped on to a virtual array of processors by using the adjacency characteristic of the input e.g. alphabets and numbers would have their inherent adjacency characteristics. Similarly images would have the frequency bands, intensity, and spatial coordinates as the adjacency characteristics per pixel etc. For a reference pattern domain  $R$ , the GN array represents all possible combinations of  $P$  in  $R$ . Each GN node is initialised with a distinct pair  $p$  from the input domain  $R$ . Also, each GN node executes an instance of the full GN code. A GN instance keeps a record of the number of times it encounters a matching input pair within its bias array. Each row of the bias array comprises a list of the adjacent GNs relating to a matched input pair. The bias row counter is incremented for every new pair encountered by the instance. A new pair is defined as the one which has a different set of adjacent GN nodes to the existing rows of the bias. The GN algorithm may be categorised into the following three stages:



**Fig. 1.** A simplified representation of the GN array’s store (memorise) and recall operations



**Fig. 2.** Structuring WSN with a Chord overlay

1. Mapping of the input pattern to the appropriate GN nodes.
2. Marking the end of the incoming pattern.
3. Bias update and lookup operations for pattern recall or memorisation.

Most of the steps within these stages may be executed in parallel. Stage 3 operation will result in one of the two possible outputs, namely memorisation or recall. A memorisation process is initiated if an input pattern is not matched with stored patterns. On the other hand, a recall is the result of a match being found amongst the stored patterns [2][5]. The GN algorithm requires that the nodes are setup as an ordered array where each node is aware of its adjacent nodes. In our proposed scheme, the Chord lookup protocol will be used to structure the WSN, as can be seen in Figure 2.

### 3 A Structured In-Network Processing Scheme for WSNs

Implementation of the GN code on an unstructured and dynamic WSN is not feasible since the GN algorithm assumes that the network be deterministic to support its real-time processing requirements. Also, the GN algorithm requires a structured network for gathering node adjacency information [2]. Our proposed scheme ensures that a suitable structure is appropriated to the WSN (using the Chord protocol) to meet these requirements.

Traditional WSN applications may constantly get interrupted owing to nodes' join, leave, and failure. The Chord protocol in our proposal defines a distributed lookup mechanism in which the position fields of the GN pairs and WSN nodes are used to provide a robust self-organising overlay (*virtual topology*) capable of handling the WSN dynamics. In addition, the overlay provides a reliable mechanism [4] for locating the WSN node designated for hosting a particular GN pair. The overlay uses an in-network distributed hash table (DHT)-based mechanism to avoid relying on a centralised entity for content discovery and structure maintenance.

In our scheme, every WSN node is assigned a key called the *WSN\_key* generated from its position in the network in our scheme. WSN node's location discovery is beyond the scope of this paper, however, [6] [7] propose effective distributed techniques for WSN node position discovery. An input pattern  $P$  of size  $N$  arriving at a node say  $X$  of the GN array may be visualised as a collection of  $N \times p(\text{value, position})$  pairs. On receiving the input pattern  $P$ , node  $X$  decomposes  $P$  into  $N$  input pairs. A new *GN\_key* is generated for each input pair  $p$  using its position field's value. Node  $X$  next looks up its local memory for any existing *GN\_keys* generated from previous inputs that can be reused for the current input pattern. All additional *GN\_keys*, as required, are generated. Subsequently, all generated *GN\_keys* are mapped by node  $X$  to corresponding WSN nodes based on the individual *WSN\_key* ranges of each WSN node. Both the *GN\_keys* as well as the *WSN\_keys* are generated using one of two hash functions supported by the Chord protocol, namely, SHA-1 or MD-5. All WSN nodes including node  $X$  which will be involved in the pattern recognition operation would

forward the incoming GN pairs from node  $X$  to their respective GN instances in the array. Finally, the GN instances running on the participating WSN nodes will collaboratively complete the pattern recognition process.

The structured WSN topology in our scheme ensures that all search operations for finding adjacent GN pairs are completed in  $O(\log n)$  time [4]. The network layer handles the data communication and routing aspects. The separation between the network layer and the Chord layer allows for flexibility and efficiency in selecting the optimal routes based on routing techniques described in [8]. We assume in our design that all active WSN nodes, regardless of their contribution to the pattern recognition process, participate in routing.

All WSN nodes participating in the pattern recognition process are arranged, by the Chord protocol, in a circular structure called the identifier circle - based on their respective  $WSN\_keys$ . The identifier circle is divided into value ranges based on  $WSN\_key$  values. Each WSN node in the identifier circle is responsible for a range of  $WSN\_keys$  starting from its own  $WSN\_key$ . GN pairs are distributed among the WSN nodes based on the individual  $GN\_keys$  of the pairs. The WSN node responsible for hosting a GN pair is located on the identifier circle by checking for the existence of the  $GN\_key$  (for the GN pair) within the  $WSN\_key$  range of the WSN node. These steps are shown as a pseudo code in Figure 3.

The total number of available nodes in the WSN must be equal to or greater than the size of the input pattern to ensure that every GN pair is mapped to its designated WSN node. If the required number of WSN nodes is less than the input pattern size, then the existing WSN nodes cater for missing nodes by hosting multiple GN pairs derived from the input pattern. The consistent hashing mechanism of Chord would to some extent balance the load resulting from the handling of multiple GN pairs per node in this case.

```

1  for i = 1 to N
2    for j = 1 to N
3      if  $GN\_Key_i \in \text{range}(WSN\_Key_j)$ 
4        GN pairi is mapped to WSN nodej
5  for available WSN nodes  $\geq N$ , each GN pair will be mapped to a
6  unique WSN node.
7  Where N is the size of the arriving pattern P
8   $\text{range}(WSN\_Key_j) = (\text{predecessor's } WSN\_key, WSN\_key_j]$ 

```

Fig. 3. Pseudo code for mapping GN pairs to WSN nodes

## 4 Maintaining the Virtual Topology

In [9] [10], Zou et al refer to the relationship between nodes in the identifier circle as links and classify Chord's links into short links and long links. Nodes adjacent to any given node within the identifier circle are called its successor and predecessor nodes respectively. Long links are introduced to provide short cuts within

the identifier circle. Each node participating in Chord creates a routing table called the *finger\_table* containing these long links as its entries. All *finger\_tables* contain a maximum of  $k$  unique long links, where  $k \leq m$ , where  $m$  is the number of entries in the *finger\_table* [4]. Short and long links are prone to disruptions owing to the dynamic nature of WSNs. Therefore, the process of maintaining and verifying the correctness of these links is essential for maintaining the *virtual topology*. The Chord protocol implements three different types of periodic updates for maintaining the *virtual topology* - *stabilize*, *notify* and *fix\_fingers*. The *stabilize* and *notify* updates are used for learning about newly joined WSN nodes and node failures in short links. Whereas the *fix\_fingers* update is used to adjust the long links by updating nodes' *finger\_tables*.

A new WSN node discovers its location in the *virtual topology* after communicating with an existing WSN node in the Chord's identifier circle. Using its *WSN\_key* the node successfully finds its location in the identifier circle and updates its adjacent nodes with its position. Subsequently, the newly joined node contacts its successor, i.e. the first node with a higher ID than this node, to acquire all *GN\_keys* of the GN pairs that belong to it along with the GN pairs falling in its *WSN\_key* range on the identifier circle. When leaving the *virtual topology*, a node migrates its GN pairs and its *GN\_keys* to its successor in the identifier circle. The consistency of the identifier circle is preserved by ensuring that a leaving node informs its successor prior to leaving and requests a *stabilize* update routine to be executed by it.

## 5 Performance Evaluation and Results

The GN algorithm using a Chord overlay for the WSN was simulated using a Java DHT simulator [11]. We assume that all WSN nodes are placed within one hop from each other. Consequently, the network layer routing schemes do not affect data collected from the Chord layer. The simulations implement the application design described in Section 3. The GN pairs of the decomposed input pattern are relayed to their designated WSN nodes by matching their individual *GN\_keys* with *WSN\_key* ranges of the WSN nodes.

### 5.1 Simulation Parameters

The *finger\_table\_entry* parameter depicts the number of entries in the *finger\_table* of a WSN node that are actually in use. The variations in the average path length owing to the changes made to the following parameters were studied:

- *finger\_table\_entries* in actual use,
- network size, and
- the percentage of network nodes with updated *finger\_tables*.

Chord suggests using all  $m$  entries of the *finger\_table* to achieve optimal lookup, where  $m$  is the number of bits used in the hashing process. This means that  $m$  entries of the table need to be updated regularly to reflect the current

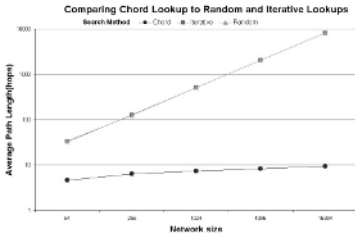
network status and the sensor network needs to dedicate a significant portion of its scarce memory and processing resources to verify the correctness of the *finger\_table*. A smaller number of the *finger\_table\_entries* may lead to sub-optimal lookups, but would make better use of the scarce resources. The effects of varying *finger\_table\_entries* on the average path length were studied as part of the simulation. Both the *finger\_table\_entry* parameter as well as the network size were varied in our study. Several simulations were performed with different percentages of WSN nodes getting their respective *finger\_tables* updated. Nodes were randomly selected for *finger\_table* updates in this case. The average path length, for finding a match, was studied by varying the number of WSN nodes randomly selected for updates.

## 5.2 Analysis

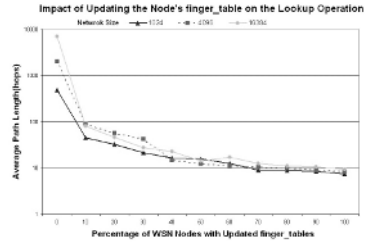
The result of the simulation for evaluating three most commonly used lookup strategies are shown in Figure 4. The Figure shows the average path length for random, iterative, and Chord key lookups being performed over network size ranging from 64 to 16384 nodes. The iterative lookup sequentially traverses a random set of WSN nodes and hence achieves approximately the same average path length as the random lookup. When compared with iterative and random lookups, Chord markedly outperforms both of these since it takes advantage of the routing information present in its *finger\_table*. Average path lengths for Chord stay sub-linear with increasing network sizes and hence provide much better scalability than random and iterative lookups.

As can be seen from Figure 5, the average path length shows a consistent trend over all three simulated network sizes. In each case, the average path length significantly drops after 10% of the network has been updated, thus indicating that the lookup performance can tolerate sub-optimal network updates. This result suggests that *finger\_table* updates can be done more selectively and less frequently in the network without significantly increasing the average path length value. Our 32-bit Chord identifiers were derived from the hash of the WSN node locations. Hence, each WSN node was assigned a *finger\_table* with 32 entries initially. The *finger\_table\_entry* parameter was varied between 1 to 32 for network sizes of 1024, 4096 and 16384 to study its effect on the average path lengths. The lookup cost for all the networks stayed consistently high for all *finger\_table\_entry* values less than 20. The average path length showed a sudden drop for *finger\_table\_entry* values greater than 20. These results are shown in Figure 6.

Smaller values of the *finger\_table\_entry* parameter lead to more number of hops for the WSN node search operation. As the value of the *finger\_table\_entry* parameter is increased, the number of hops required reduces sharply - from size 20 onwards. Smaller *finger\_table\_entry* values imply that all WSN nodes will simply forward the incoming GN pair keys to their respective successor nodes if the key doesn't belong to their individual ranges. Thus, a sequential traversal of the identifier circle results leads to higher average path lengths. In light of these findings, *finger\_table\_entry* values up to 20 showed limited or no response to the updates. For greater *finger\_table\_entry* values, the average path length was sig-



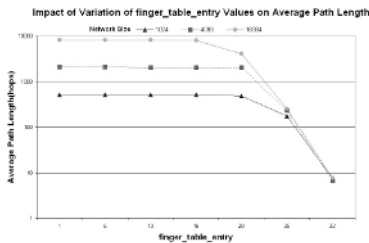
**Fig. 4.** Average path lengths for random, iterative, and Chord lookup strategies over network sizes ranging from 64 to 16,384 nodes



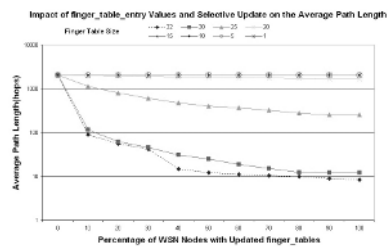
**Fig. 5.** Average path lengths for percentage nodes selectively updated over 1024, 4096, and 16,384 node networks

nificantly reduced for node update proportions greater than 10%. The proposed overlay design therefore needs to strike an optimal trade-off between the lookup speed and the corresponding maintenance overhead for the Chord layer. The selective update method was repeated over various *finger\_table\_entry* values.

The changes in network sizes, as applied in Figure 5 and Figure 6, did not show any significant change to average path length values. We therefore took a constant network size of 4096 for studying the impact of varying the percentage of selective *finger\_table* updates on the average path length. Figure 7 shows the optimal trade-off occurring whilst using a sub-optimal Chord's configuration of *finger\_table\_entry* value of 25. It may be seen from this Figure that, the average path length for *finger\_table\_entry* value of 25 remains almost constant as the total numbers of WSN nodes with updated *finger\_tables* is increased. This implies that having more nodes updated does not significantly improve the average lookup path length. Also, the average path length stays significantly lower than those for *finger\_table\_entry* values < 20. It may be pointed out that the above results apply to cases with the *finger\_table* set to 32 ( $m=32$ ).



**Fig. 6.** The effect of varying the *finger\_table\_entry* on average path length for 1024, 4096, and 16384 node networks



**Fig. 7.** *finger\_table\_entry* variations with selective update versus average path length for the 4096 node network

## 6 Memory and Processing Requirements for Structured WSNs

WSN nodes are usually inexpensive battery-operated devices with low computation and communication capabilities. A typical smartdust sensor node has the following specifications [12]:

```
CPU 8-bit, 4MHz
Storage 8K Instruction flash
512 byte RAM 512 byte EEPROM
Operating System: TinyOS
OS code space 3500 bytes
Available code space 4500 bytes
```

The TinyOS operating system itself occupies 4Kbytes of the flash memory leaving 4Kbytes of storage for the GN and the Chord codes. The *finger\_table* has a depth of  $m$  rows with each row containing two entries of length  $m$  - hash value of the successor node and index of the node. The GN instance on each WSN node was implemented with a maximum of three *bias\_arrays*. The *bias\_array* size was set to 10 for these simulations. The *key\_list* entries were the hashed values of the value-position pairs of the input pattern comprising a character string.

The cost of storage for the GN instance and the Chord protocol on a single WSN node may thus be calculated as follows.

$$Cost_{storage} = \textit{finger\_table storage} + \textit{key\_list storage} + \textit{bias\_array storage} \quad (1)$$

where,

$$\textit{finger\_table storage} = m * (m + m) \quad (2)$$

$$\textit{key\_list} = 2 * P * m \textit{ bits}, \quad (3)$$

$P$  is the number of pairs in the input pattern

$$\textit{bias\_array} = 3B \textit{ bits}, \quad (4)$$

$B$  is the length of the *bias\_ array* = 10

$$Cost_{storage} = 2m^2 + 2Pm + 3B \textit{ bits} \quad (5)$$

$$Cost_{storage} \cong 2m^2 \textit{ bits} \quad (6)$$

Assuming that the 2Kbytes of memory available for the executables is used in its entirety, the remaining 2Kbytes of memory can be used for storing the *finger\_table*, *key\_list*, and *bias\_array* of the GN instance running over a WSN node.

$$2m^2 \leq 2 * 8 \textit{ Kbits} \quad (7)$$

$$m \leq 89 \textit{ bits} \quad (8)$$

It may be seen from inequality 8 that the maximum hash key length used for generating both *WSN\_keys* and *GN\_keys* cannot be greater than 89 bits. Assuming unique identifiers (keys) for each WSN node, it can be deduced that the maximum possible combination of unique keys for the WSN nodes that may participate in the GN pattern recognition process is equal to  $2^{89}$ . This upper bound also determines the maximum length of input patterns that may be stored by the GN array, in the case where each input value, position pair is being mapped to a unique WSN node.

The total cost of computation on each WSN node comprises a summation of the key generation, *finger\_table* lookup/update, and the bias lookup/update costs. All three operations defined above are hash calculations. Hence the total cost of computation, in the worst case, based on the hash computation metrics from [13] may be estimated as follows.

$$\begin{aligned}
 Cost_{computation} &\cong key\_generate + finger\_table & (9) \\
 &lookup/update + bias\ lookup/update \\
 &\cong 1080\ \mu sec + m * 1\ \mu sec + 1\ \mu sec * B \\
 &\cong 1179\ \mu sec\ for\ an\ 89\text{-bit}\ hash\ key\ and\ a\ 10\ entry \\
 &bias\ table\ hash\ computation.
 \end{aligned}$$

The pattern analysing experiment for purposes of this paper assumes non-overlapping input patterns for the GN nodes and therefore, at any given time, a sensor node will be performing a single hash operation. In our experiment, the *finger\_table* size,  $m$ , is taken as 32. Therefore, from equations 6 and 9, the costs for storage and computation for the experiment are 256 bytes and  $1122\ \mu sec$ s, respectively. The storage cost for the experiment is well below the maximum storage capacity of 2 Kbytes of a smartdust sensor node, and the cost of computation per node is fairly small and it could thus be safely neglected.

## 7 Conclusions

We have presented a self-organising scheme, called the *virtual topology*, for imparting a structure to an otherwise unstructured WSN. Chord's adaptive circular structure was proposed to manage the dynamics of the network. The Chord overlay decouples the application (GN) layer from the physical network uncertainties and provides a deterministic virtual environment for supporting the real-time requirements of the application. The GN array and our *virtual topology* utilise the sensor network in a decentralised and balanced manner for performing in-network computations. The simulation results clearly indicate that the deployment of the *virtual topology* for the pattern recognition application on the WSN is feasible both in terms of the memory usage and the computational requirements of a WSN node.



## References

1. Chee-Yee, C., Kumar, S.: Sensor networks: evolution, opportunities, and challenges. *Proceedings of the IEEE* **91** (2003) 1247–1256
2. Khan, A.I.: A peer-to-peer associative memory network for intelligent information systems. In: *The Thirteenth Australasian Conference on Information Systems*. Volume 1. (2002)
3. Khan, A.I., Mihailescu, P.: Parallel pattern recognition computations within a wireless sensor network. In: *ICPR*. Volume 1. (2004) 777–780
4. Stoica, I., Morris, R., Liben-Nowell, D., Karger, D., Kaashoek, M.F., Dabek, F., Balakrishnan, H.: Chord: a scalable peer-to-peer lookup protocol for internet applications. *IEEE/ACM Transactions on Networking* **11** (2003) 17–32
5. Khan, A.I., Isreb, M., Spindler, R.: A parallel distributed application of the wireless sensor network. In: *Seventh International Conference on High Performance Computing and Grid in Asia Pacific Region*. (2004)
6. Iyengar, R., Sikdar, B.: Scalable and distributed gps free positioning for sensor networks. In: *ICC '03*. Volume 1., IEEE (2003) 338–342
7. Savarese, C., Rabaey, J.M., Langendoen, K.: Robust positioning algorithms for distributed ad-hoc wireless sensor networks source. In: *the General Track: 2002 USENIX Annual*. (2002) 317–327
8. Qiangfeng, J., Manivannan, D.: Routing protocols for sensor networks. In: *CCNC 2004*. (2004) 93–98
9. Zou, F., Chen, S., Ma, F., Zhang, L., Tang, J.: Using the linking model to understand the performance of dht routing algorithms. *ISPA 2004* (2004) 544–549
10. Zou, F., Zhang, L., Li, Y., Ma, F.: Effect of links on dht routing algorithms. In: *GCC*. Volume 3032/2004., Springer-Verlag GmbH (2004) 308–315
11. Mao, Y.: Dht java simulator. <http://www.cis.upenn.edu/macy/research/> (2004)
12. Perrig, A., Tygar, J.: *Secure Broadcast Communication in Wired and Wireless Networks*. Kluwer Academic Publishers (2002)
13. Ganesan, P., Venugopalan, R., Peddabachagari, P., Dean, A., Mueller, F., Sichertiu, M.: Analyzing and modeling encryption overhead for sensor network nodes. In: *the 2nd ACM international conference on Wireless sensor networks and applications*. (2003)

# Building Graphical Model Based System in Sensor Networks

Dongyu Shi, Jinyuan You, and Zhengwei Qi

Department of Computer Science and Engineering Shanghai Jiao Tong,  
University, Shanghai 200030, P.R. China  
sshidy@yahoo.com.cn, {you-jy, qi-zw}@cs.sjtu.edu.cn

**Abstract.** Consisting of a large number of sensing and computational devices distributed in an environment, a sensor network can gather and process data about a physical area in real time. Due to the limited computing power in each sensor, limited bandwidth connections, limited storage and other limitations, how to deal with the data and uncertainty knowledge is one of the most important and central problems in such kind of distributed systems. This paper presents a graphical model based intelligent system that can model the uncertainty knowledge in sensor networks. This system uses belief messages as a basis for communication. We focus on parameter learning process for building the model, and experiments are presented.

## 1 Introduction

Advances in computing and communication over wired and wireless networks have resulted in many pervasive distributed computing environments, such as Internet, local area networks, ad hoc wireless networks, and sensor networks. These environments often come with different distributed sources of data and computation. Sensor networks are a new kind of distributed system, which are fast developing in recent years. They can sense certain phenomena in an environment, while gathering data in real time for further analysis. They consist of a large number of low-cost, low-power multifunctional computational devices that can be easily deployed in the environment.

Sensor networks are typically used in one of the two modes of operation: either the data from the sensors are extracted from the network and shipped to a server for offline processing; or the information obtained from the sensors is aggregated using local operations in real time within the network itself. How to deal with the “raw” data is one of the central questions in sensor network. Since the bandwidth connections are limited, the extraction of complete data sets can be very expensive, requiring large amounts of communication that drains the energy of these devices. So it is very attractive for mining knowledge from “raw” data locally to reduce communication. But when the data processed online within the network, what form should the information take? How we can compute the uncertainty knowledge from the data? And how should we organize the overall flow of information in a distributed fashion?

In this paper we use a probabilistic graphical model based intelligent system to solve these problems. Graphical models have become increasingly popular as means to structure uncertainty knowledge in complex domains and thus to facilitate reasoning in such domains. Bayesian Networks and Markov networks are the two most popular kinds of graphical models that have been widely used in uncertainty problems. In many applications, the main task is to form beliefs about the state of the distributed network system based on the collected sensor data. There are some works [5,7] that have been done in this area, and their model has been proved very effective in uncertainty knowledge representation. Here we focus on the parameter learning process for building the distributed intelligent system.

The rest of the paper is organized as follows: we begin in Section 2 with some background and related work. In Section 3, we describe in detail our graphical model based system in a sensor network. In Section 4, we provide the parameter learning process in the system. Section 5 shows the experimental results and the conclusion.

## 2 Background and Related Work

In this section we first give a brief review of probabilistic graphical models, and then introduce the sensor networks and related works about modeling and mining data in sensor networks and other distributed systems.

### 2.1 Graphical Models

A graphical model is a family of probability distributions defined in terms of a directed or undirected graph. The nodes in the graph are identified with random variables, and the (lack of) arcs represent conditional independence assumptions. Hence they provide a compact representation of joint probability distributions [4].

Undirected graphical models are also called Markov networks [8]. They have a simple definition of independence: two (sets of) nodes  $A$  and  $B$  are conditionally independent given a third set  $C$ , if all paths between the nodes in  $A$  and  $B$  are separated by a node in  $C$ . By contrast, directed graphical models are also called Bayesian Networks (BNs). They are directed acyclic graphs. They have a more complicated notion of independence, which takes into account the directionality of the arcs. Bayesian network shows the causal relations among its variables. Once completed, both Bayesian network and Markov network can be used to derive the posterior probability distribution of one or more variables using an inference process, with the observed particular values for other variables in the network, or to update previous conclusions when new evidence reaches the system.

### 2.2 Sensor Networks and Distributed Data Processing

The widespread distribution and availability of small-scale sensors, actuators, and embedded processors is transforming the physical world into a computing platform. Sensor networks that combine physical sensing capabilities such as temperature, light, or seismic sensors with networking and computation capabilities will soon become ubiquitous. Applications range from environmental control, warehouse inventory, and health care to scientific and military scenarios.

Sensor networks are much more tightly integrated with the environment than previous distributed systems. Instead of relying on a small number of interfaces, every (sensor) node in the system is embedded with and in contact with the environment. On the other hand, they rely fundamentally on computation being done by a large number of the distributed nodes. The sensor nodes are generally low-cost, low-power multi-functional devices that can be easily deployed in the environment. Algorithms for these nodes must be implemented very cheaply. Since communication in today's networks is orders of magnitude more expensive than local computation, it is necessary to use in-network storage and processing to vastly reduce resource usage and extend the lifetime of a sensor network.

Sensor networks naturally have much uncertainty knowledge in their systems. The application of distributed data mining and computation to the system is very important, and becomes a core task. Some works have been done in this area. In [6], Carlos Guestrin etc. present distributed regression as an efficient and general framework for in-network modeling of sensor data. In [5], C. Crick and A. Pfeffer use Loopy Belief Propagation as a basis for communication in sensor networks. In [7], M.A. Paskin and Guestrin discussed robustness of probabilistic inference in distributed systems, especially in sensor networks. There are also some works about using probabilistic networks in distributed data mining. In [10], [3], K. Sivakumar and R. Chen discussed Bayesian Network structure learning from distributed data. And in [2], they provide a new algorithm for learning parameters of a BN in a distributed way.

### 3 Modeling Data in a Sensor Network as a Graphical Model

There is a great deal of uncertainty in sensor network systems. Signals detected at physical sensors have inherent uncertainty, and they may contain noise from the environment. Sensor malfunction might generate inaccurate data, and unfortunate sensor placement (such as a temperature sensor directly next to the air conditioner) might bias individual readings. Reasoning under uncertainty to form coherent beliefs is a major task in sensor network systems.

Here we build a graphical model to deal with the task in a simulated "Fire-Detection" sensor network. It is supposed to detect fire from sensors collecting temperature and light in a wide area. Sensor networks and sensor nodes have many limitations [1], and one of the general proposed design is the Hourglass architecture [9]. It envisions 4 kinds of nodes: sensor nodes, data nodes, communication nodes and processing nodes. In our model we consider three kinds of nodes:

- Sensor nodes are deployed in the whole target area and responsible for collecting and storing raw sensed data. Each sensor individually provides a reading for a state variable at a particular point. The sensor properties may also affect the reading. Sensor nodes only communicate with the local processing node.
- Processing nodes perform some computation on the data within a local network. Here we divide the whole sensor area into a series of sub-area. In each sub-area there's a local network with a processing node examining a set of sensor nodes. It should form beliefs about high-level variables (such as fire, temperature and light in the sub-area) from sensor readings. The communication between sub-areas will be

performed by the neighbor processing nodes. And finally, the network of the processing nodes will send information to the central management node.

- The central management node deals with central tasks.

Fig.1 shows an example of a sub-area network. The *Fire*, *Temp* and *Light* are high-level variables that need to be compute in a processing node. The others are local variables. The “SS1” means variables related to local SenSor node 1, similar with “SSn”. We model the sub sensor network as a Bayesian network. The directed edges show the dependent relationships between variables. The high-level variables in adjacent areas are highly correlated. We use a Markov network to represent the relationships between high-level variables in adjacent processing nodes.

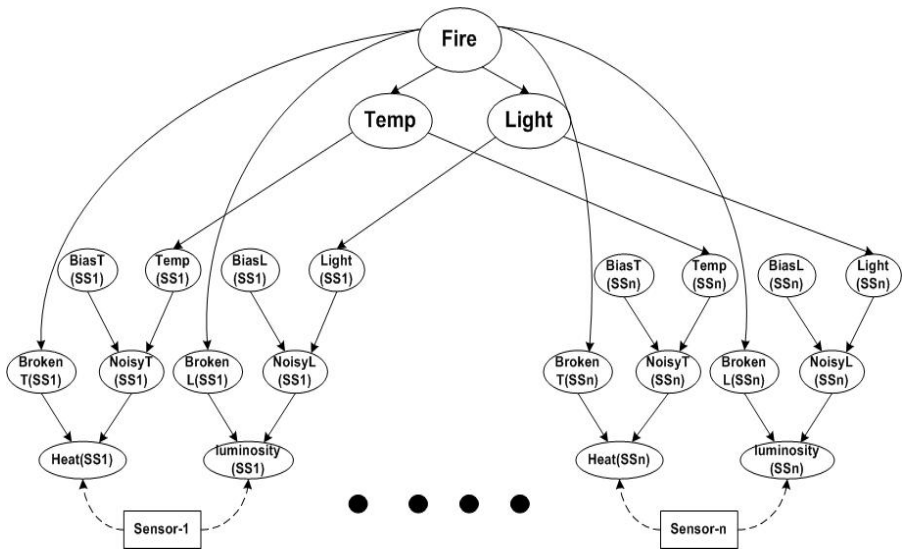


Fig. 1. A local Bayesian network with a processing node and n sensor nodes

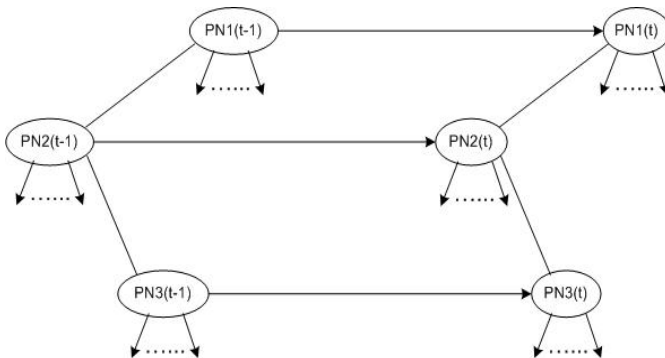


Fig. 2. The dynamic model of the state change between time t-1 and t

Since the environment is changing by time, the state of a sensor network will change dynamically. Here we propose a dynamic model for the whole system. Fig. 2 shows the dynamic model of the communication between processing nodes. The undirected edges between processing nodes (PN1, PN2...) compose a Markov network. There is a directed edge from each processing node at time point  $t-1$  to the corresponding processing node at time  $t$ . So the whole model reveals the state change of the hybrid graphical model between time  $t-1$  and  $t$ .

## 4 Parameter Learning Process

Learning parameters of a probabilistic graphical model in a distributed way is one of the major topics in Distributed Data Mining. How to apply the learning process in a sensor network environment can be a critical technique for solving complicated uncertainty problems. In this model, we must first decide the parameters of the distributed graphical model before we make inference and send results to central nodes. Other than arbitrarily decide the parameters, we generate a set of training data samples, and try to learn them within the network.

Since data is distributed among different sensors, we present a collective strategy to learn the parameters in the hybrid model with local Bayesian networks and high-level Markov networks. The primary steps are in the following:

### 4.1 Learning Parameters of Local BN

At each processing node in a sub-area, learn the variables involved in local BN model based on local data set. We can use the Maximum Likelihood (ML) method to learn the network parameters. And the likelihood function is as follows:

$$L(\theta : D) = \prod_m P(x_1[m], \dots, x_n[m] : \theta) \quad (1)$$

$$= \prod_i \prod_m P(x_i[m] / Pa_i[m] : \theta_i). \quad (2)$$

$m = 1 \dots M$  is the number of sample data sets. Taking Fig. 1 as an example, a complete sample data set includes *Heat(SS<sub>*i*</sub>)* and *luminosity(SS<sub>*i*</sub>)* from readings of sensor node  $i$ , *Temp(SS<sub>*i*</sub>)* and *Light(SS<sub>*i*</sub>)* from measuring the state of sensor node  $i$ , assessment of *Bias(SS<sub>*i*</sub>)*, setting of *BrokenT(SS<sub>*i*</sub>)* and *BrokenL(SS<sub>*i*</sub>)*, all with  $i$  from 1 to  $n$ ; and *Temp*, *Light*, *Fire* from measuring the state of the sub-area. We put all the sample data sets into Equation. 2 according to the network structure of parent-child relationships ( $P(\text{child}/\text{Parent})$ ), and learning the parameters of the local BN offline.

### 4.2 Select Samples for Parameter Learning of High-Level Markov Network

At each processing node  $PN_k$ , based on the local BN, we can compute the joint distribution of the high-level variables. Based on this joint distribution, a subset of samples is selected. Let  $Icol_k$  denote the set of indices of these samples. Then we use the samples to learn parameters of the Markov network among high-level variables of each local BN.

Here we first discuss the sample selection process. Let  $l_A(\cdot)$  and  $l_B(\cdot)$  denote the estimated likelihood function involving the local variables at neighbor sub-area  $A$  and  $B$  respectively. The observations at each local network are ranked based on how well it fits the local model using the local likelihood functions. The observations at  $A$  with large likelihood under  $l_A(\cdot)$  are evidence of “local relationships” between variables in  $A$ , whereas those with low likelihood are possible evidence of “cross relationships” between variables across sub-networks. Let  $HL$  and  $LL$  denote high-level variable sets and sets of other variables in a local network respectively. Consider the marginal distribution of  $P(HL) = \sum_{LL} P(HL, LL)$ . Therefore, if a sample configuration for  $HL$  variables has low likelihood value under  $P(HL)$ , then there is at least one configuration of  $LL$  variables such that the corresponding joint configuration  $(HL, LL)$  has low likelihood value under  $P(HL, LL)$ .

We set a threshold  $T$  under  $P(HL)$  for each local network and choose the samples whose likelihood is lower than this threshold. As discussed above, these samples are the main part of samples whose likelihood is small under  $P(HL, LL)$ . Then we introduce a random sampling for the samples whose likelihood is higher than the threshold. We put the two parts of selected sample data sets together, and put them into the high-level Markov network parameter learning. The reason why we do not choose samples under  $P(HL, LL)$  is that the total number of configurations  $(HL, LL)$  is prohibitively large, but a node in a sensor network has limited memory and computing power.

### 4.3 Learning Parameters of High-Level Markov Network

As the whole network structure is known, and the samples in every local network are chosen, theoretically we can transmit all the samples to central management node and obtain the values of each variable which maximizes the likelihood of the samples. But due to the limited bandwidth connection of sensor network, it will not be feasible. Besides, if the number of sensors is very large, the computation can be a heavy burden for the central node. So we consider learning the parameters within the network. Since we model the network among processing nodes a Markov network, if it has no cliques (fully connected graphs) containing more than 2 nodes, the joint probability distribution over the states of all processing nodes can be decomposed into the product of pair-wise interactions between adjacent nodes. And then we can just exchange local samples between adjacent processing nodes, and use Maximum Likelihood method to learn the parameters locally by processing nodes. Assuming that the number of total selected sets is  $Selec$ , the likelihood function of the whole Markov network is as follows:

$$L(\theta : D) = \frac{1}{Z} \prod_{m=1}^{Selec} \psi(HL_1[m], \dots, HL_n[m] : \theta) \tag{3}$$

$$= \frac{1}{Z} \prod_j \prod_{m=1}^{Selec} \psi_j(HL_{j_1}[m], HL_{j_2}[m] : \theta_j) . \tag{4}$$

$m = 1 \dots Selec$  is the number of sample data sets;  $j_1$  and  $j_2$  are adjacent nodes that are connected by edge  $j$ ;  $Z$  is a normalization constant. Taking Fig. 2 as an example, the

clique functions  $\prod_{m=1}^{Selec} \psi_1(HL_1[m], HL_2[m])$  and  $\prod_{m=1}^{Selec} \psi_2(HL_2[m], HL_3[m])$  will be included in Equation.4.

If we carefully design the network structure, we can apply this method to the system. In fact in a sensor network, any 3 processing nodes being fully connected may not be necessary. Even when there are cliques containing 3 or more nodes, we can use the above decomposition method as an approximation, and it tend to be a good approximation.

Combining the local BN models and the high-level Markov network with all the parameters, the whole hybrid graphical model can be built. After all these work, we already obtain all the parameters of the network structure. But this is only a snapshot model at a certain time. To obtain a dynamic model we should compute high-level parameter changing in processing nodes between two time slices, and that is the probability  $P(HL_i^{(t)} / HL_i^{(t-1)})$ . Here we take the assumption that it is a Markov process, the state at  $t$  is only affected by the state at  $t-1$ . We continuously observe the state changing in different periods, and collect data to obtain the probability distribution.

Until now the whole intelligent system for the sensor network has been built. We can apply probabilistic inference techniques to this system, sending beliefs rather than raw data in the sensor network, and get “smart” results of the state in the area that the sensors deployed. We should notice that the probabilistic inference problem in a sensor network environment has already been discussed in [5] by C. Crick and A. Pfeffer. Although our system is different in some kind (like dynamic property), and the inference process requires some further analysis, the feasibility and efficiency are showed in the experiment.

### 5 Experiments and Conclusions

In our first experiment, we simulate a sensor network of 3 processing nodes and 14 sensor nodes in networks. To test the effectiveness of our parameter learning method, we generate 500 sets of sample data. We first compute the parameters in a central way

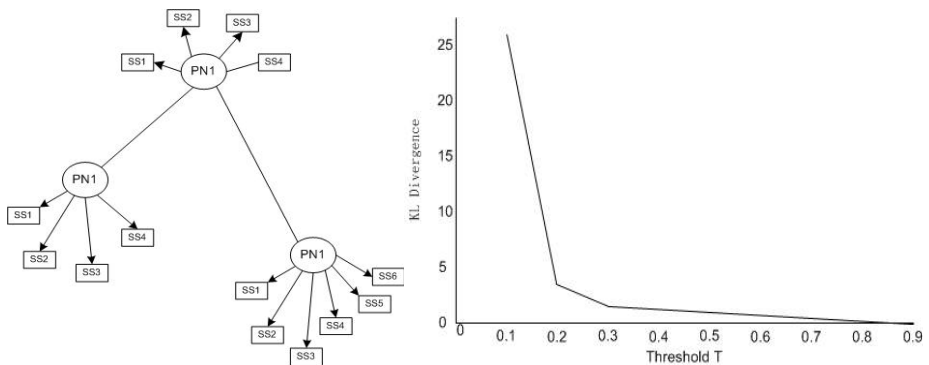


Fig. 3. The network structure and KL-divergence between two joint probability distributions



using Maximum Likelihood method, and then compare them to the parameters we learned in the distributed way described in section 4. Fig. 3 shows the network structure and the Kullback-Leibler Divergence (or KL distance) between the two joint probability distributions as the threshold  $T$  in the second step of the learning process increases.

In another experiment there are 8 local networks, each with a processing node and 3 to 7 sensor nodes, collecting data from more real world environment. The structure of processing nodes is show in Fig.4. The KL divergence between the probability distributions estimated based on our distributed parameter learning method and the parameter obtained using a centralized ML approach is computed. In particular, we illustrate the results for the probability distributions at two different nodes PN3 and PN6. Take PN6 as an example, we compute the sum over all the possible values of its neighbor nodes, of the KL distance of the probability distribution between distributed approach and central approach. The expression is as follows:

$$\sum_{HL_5, HL_7, HL_8} KL(P_{ctr}(HL_6), P_{dstr}(HL_6)). \tag{5}$$

$$P_{ctr}(HL_6) = \frac{1}{Z_{ctr}} \psi_{ctr}(hl_5, HL_6) \psi_{ctr}(HL_6, hl_7) \psi_{ctr}(HL_6, hl_8). \tag{6}$$

$$P_{dstr}(HL_6) = \frac{1}{Z_{dstr}} \psi_{dstr}(hl_5, HL_6) \psi_{dstr}(HL_6, hl_7) \psi_{dstr}(HL_6, hl_8). \tag{7}$$

$hl_5, hl_7$  and  $hl_8$  are a set of certain values that  $HL_5, HL_7$  and  $HL_8$  take respectively.  $\psi_{ctr}$  and  $\psi_{dstr}$  are clique functions of which the parameters are learned in the central and distributed way respectively.

Fig. 5 depicts the similar KL distance for PN3 and PN6. It shows that generally the KL distance is getting smaller when the threshold getting higher, which means more data are transmitted in the network. But even with a small data communication, it is quite close to that obtained by the centralized approach.

It should be noticed that a real sensor network system could have several hundreds to thousands of sensors. If we learned parameters in a central way, it would be a heavy burden for the server (the central management node), and making it infeasible. So we do not make comparisons between these two ways in a large sensor network system. After learning the temporal parameters -  $P(HL_i^{(t)} / HL_i^{(t-1)})$ , the whole graphical model based system is build, and can be used for inference and belief propagation.

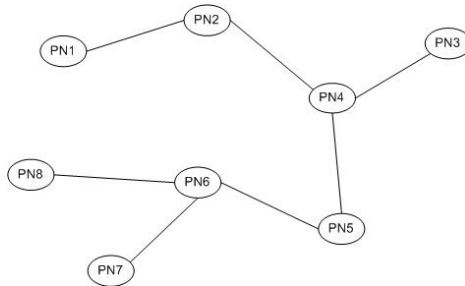
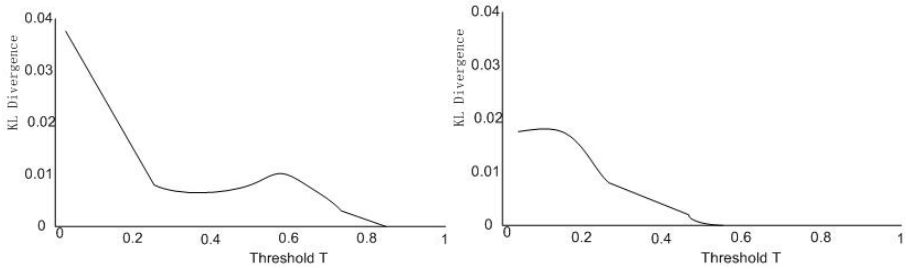


Fig. 4. The structure of processing nodes



**Fig. 5.** KL distance between probability distribution learned by our method and central approach of processing node PN3 and PN6

A few tests are run in the system. Comparing to the system in which all the data from the sensors are extracted from the network and shipped to a server for offline processing, our system mainly has advantages in the following two aspects: first, it shows a much quicker response. When there is a simulated fire, it averagely spends 1/3 to 1/2 of time for the GM-based system reporting a fire with the possibility more than 50%. This is because that the GM-based system's response has combined several sensors' readings other than one sensor's. Second, the communication is greatly reduced. We observe that the network delay is much less than raw-data-gathering system. Since the bandwidth is limited, and different local networks have different bandwidth, this quality can greatly improve the performance.

As a conclusion we present an intelligent system that can model the uncertainty knowledge by a dynamic graphical model in sensor networks in this paper. The system uses belief messages as a basis for communication. The parameter learning process for building the model is provided in detail. Experiments have shown that the distributed learning techniques are efficient, and the whole system is feasible and effective. In the future we will apply the system to more large sensor networks.

## Acknowledgements

This paper is supported by the Shanghai Science and Technology Development Foundation under Grant No. 03DZ15027 and the National Natural Science Foundation of China under Grant No. 60173033.

## References

1. Akyildiz, I. F., Weilian Su, Sankarasubramaniam, Y., Cayirci, E.: A survey on sensor networks. *Communications Magazine*, IEEE Vol. 40, Issue 8, Aug. 2002Page(s):102-114
2. Chen, R., Sivakumar, K.: A new algorithm for learning parameters of a Bayesian network from distributed data. In *Proceedings of the 2002 IEEE International Conference on Data Mining*, pages 585–588, 2002
3. Chen, R., Sivakumar, K., Kargupta, H.: Collective Mining of Bayesian Networks from Distributed Heterogeneous Data. *Knowl. Inf. Syst.* 6(2): 164-187 (2004)

4. Cowell, R., Dawid, P., Lauritzen, S., Spiegel, H.D.: Probabilistic Networks and Expert Systems. Springer, 1999
5. Crick, C., Pfeffer, A.: Loopy belief propagation as a basis for communication in sensor networks. In Proc. of the 19th Conference on Uncertainty in AI (UAI-2003), 2003
6. Guestrin, C., Thibaux, R., Bodik, P., Paskin, M., Madden, S.: Distributed regression: an efficient framework for modeling sensor network data. In Proc. of Information Processing in Sensor Networks 2004 (IPSN-04), 2004
7. Guestrin, C., Paskin, M., Madden, S.: Robust Probabilistic Inference in Distributed Systems. In Proc. of the 20th Conference on Uncertainty in AI (UAI-2004), 2004
8. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan and Kaufmann, San Mateo, CA, (1988)
9. Shneidman, J., Choi, B., Seltzer, M.: Collecting data for one hundred years. Technical report, Division of Engineering and Applied Science, Harvard University, 2002
10. Sivakumar, K., Chen, R., Kargupta, H.: Learning Bayesian Network Structure from Distributed Data. In Proceedings of the 3rd SIAM International Data Mining Conference, pages 284-288, San Francisco, CA, May 2003

# Energy-Aware Broadcasting Method for Wireless Sensor Network

Cheol-Min Park<sup>1</sup>, Dae-Won Kim<sup>1</sup>, and Jun Hwang<sup>2</sup>

<sup>1</sup> Chung-Ang Univ., Bobst-Hall 5 floor, System Software Lab,  
Dept. of Computer Science & Engineering, Seoul 221, 156-756, Republic of Korea  
raphael66@korea.com, hide@sslslab.cse.cau.ac.kr

<sup>2</sup> Seoul Woman's Univ., Dept. of Information & Media,  
Seoul 126, 139-774, Republic of Korea  
hjun@swu.ac.kr

**Abstract.** The communicative behaviors in Wireless Sensor Networks(WSNs) can be characterized by two different types: routing and broadcasting. The broadcasting is used for effective route discoveries and packet delivery. A blind flooding approach for broadcasting generates many redundant transmissions. The Dominant Pruning(DP) algorithm is reduced the redundant transmissions of packets based on 2-hop neighborhood information. However, in DP(include TDP/PDP) algorithm, a particular node is frequently selected as a rebroadcasting node and its life-time is shortened. As a result, DP algorithm is insufficient in terms of the overall energy dissipation in sensor network. In this paper, we propose the algorithm based on Partial Dominant Pruning(PDP) algorithm to enhance sensor network lifetime. We compare and analyze the simulation result of our algorithm with PDP.

**Keywords:** wireless sensor network, broadcasting, network lifetime, energy-aware.

## 1 Introduction

A sensor network is composed of a large number of sensor nodes that are densely deployed either inside the phenomenon or very close to it. Each sensor node is limited in power, computational capacities, and memory size [1]. Also, as deployed sensor nodes can't replace the battery, energy efficiency is a most important factor in sensor network.

The communicative behaviors in Wireless Sensor Networks(WSNs) can be characterized by two different types: routing(node-to-sink) and broadcasting(sink-to-node or node-to-node). Broadcasting is an essential communication requirement for sink and sensor nodes. A sink node usually floods the query request to a region of all sensor nodes in a user-demand manner, asking these nodes for returning environment information. Such an application in WSNs requires a broadcasting protocol to deliver the query information from sink to all sensor nodes.

The traditional solution to the broadcasting problem is blind flooding, where each node receiving the message for the first time retransmit it to all its neighbors. Blind flooding generates many redundant transmissions. Many broadcast algorithms besides

blind flooding have been proposed [2],[3],[4],[6],[7],[8],[9], and these algorithms utilize neighborhood and/or history information to reduce redundant packets. The dominating pruning(DP) algorithm [3] is one of the promising approaches that utilize 2-hop neighborhood information to reduce redundant transmissions. Enhancements to dominant pruning have been reported by Lou and Wu [2], who describe the total dominant pruning(TDP) algorithm and the partial dominant pruning(PDP) algorithm.

DP algorithm selects forwarding node set as the optimized 1-hop nodes to cover 2-hop neighbor nodes to reduce the number of forwarding nodes. As a result, a particular node's lifetime is shortened because it is frequently selected as a rebroadcasting node, and it affects the overall network lifetime. Although DP algorithm selects forwarding node set as the optimized 1-hop node to cover 2-hop neighbor node, it cannot make the optimized routing path from a viewpoint of sensor network.

In this paper, we propose the algorithm to improve overall network lifetime as disperse the dissipation of node energy on sensor network using the node's energy information. Our proposed algorithm is improved the routing path for a viewpoint of sensor network by adding a flag bit to broadcasting messages. Our algorithm is based on the PDP algorithm and is modified the selection process of DP algorithm. Simulation results of applying this algorithm show that the proposed method in this paper has achieved better performance than the PDP algorithm in the lifetime of the network.

The rest of the paper organized as follow. Section 2 illustrates the TDP/PDP algorithm. The energy-aware broadcasting algorithm is proposed in Section 3 and simulation results are shown in Section 4. Conclusions are finally made in Section 5.

## 2 Preliminaries

### 2.1 TDP and PDP Algorithm [2]

We use a simple graph,  $G = (V, E)$ , to represent the wireless sensor network, where  $V$  represent a set of wireless mobile hosts(nodes) and  $E$  represents a set of edges(links). The network is seen as a *unit disk graph* [10], i.e., the nodes within the circle around node  $v$  (corresponding to its radio range) are considered its neighbors.

We use  $N(u)$  to represent the neighbor set of  $u$  (including  $u$ ).  $N(N(u))$  represents the neighbor set of  $N(u)$  (i.e., the set of nodes that are within 2-hops from  $u$ ). Clearly,  $\{u\} \subseteq N(u) \subseteq N(N(u))$  and if  $u \in N(v)$ , then  $N(u) \subseteq N(N(v))$ . Throughout the paper, we assume that  $u$  (sender) and  $v$  (receiver) are neighbors.

In DP algorithm, node  $v$  just needs to determine its forwarding node list  $F(u, v)$  from  $B(u, v) = N(v) - N(u)$  to cover nodes in  $U(u, v) = N(N(v)) - N(u) - N(v)$ .

In TDP Algorithm, if node  $v$  can receive a packet piggybacked with  $N(N(u))$  from node  $u$ , the 2-hop neighbor set that needs to be covered by  $v$ 's forward node list  $F(u, v)$  is reduced to  $U(u, v) = N(N(v)) - N(N(u))$ . In the PDP algorithm, no neighborhood information of the sender is piggybacked with the broadcast packet. Therefore, the deduction of  $N(N(u))$  from  $N(N(v))$  cannot be done at node  $v$ . However, unlike the DP algorithm, more nodes can be excluded from  $N(N(v))$ . These nodes are the neighbors of each node in  $N(u) \cap N(v)$ . Such a node set is donated as  $P(u, v) = N(N(v) \cap N(u))$ . Therefore, the 2-hop neighbor set  $U$  in the PDP algorithm is  $U(u, v) = N(N(v)) - N(u) - N(v) - P$ .

Both the TDP and PDP algorithm reduce the size of  $U(u, v)$  and, hence, reduce the size of  $F(u, v)$  than the original DP algorithm. But, the PDP algorithm is more cost effective, since no neighborhood information of the sender is piggybacked in the PDP during the transmission.

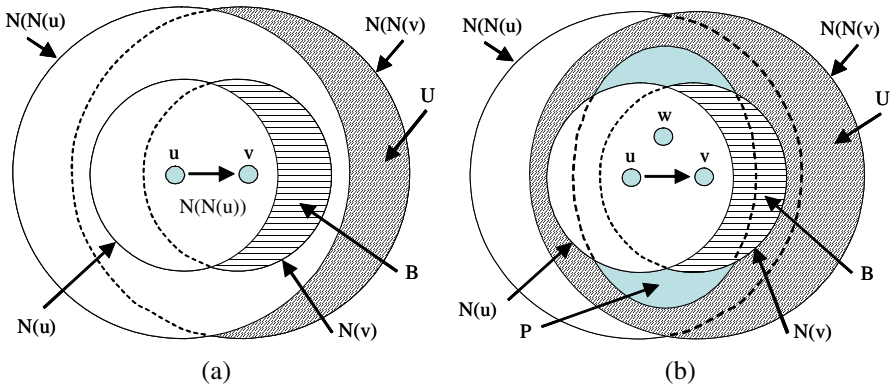


Fig. 1. Illustration for two algorithms: (a) total dominant pruning (TDP), (b) partial dominant pruning (PDP)

## 2.2 Lifetime of a Sensor Network

The definition of the lifetime of a sensor network is determined by the kind of service it provides. Hence, the lifetime of a sensor network can group into three classes.

In some cases it is necessary that all nodes stay alive as long as possible, since network quality decreases considerably as soon as one node dies. Scenarios for this case include intrusion or fire detection, and it is important to know when the first node dies.

In other cases, sensors can be placed in proximity to each other, and therefore adjacent sensors could record related or identical data. Hence, the loss of a single or few nodes does not automatically diminish the quality of service of the network. In this scenario it is needed to know the half-life period of the sensor network. Finally, for the overall lifetime of the sensor network, it is to know when the last node dies.

## 3 Energy-Aware Broadcasting Protocol

### 3.1 Basic Concepts

The PDP algorithm creates the forward node list as the optimized 1-hop nodes to cover 2-hop neighbor nodes to reduce the number of forward nodes. As a result, it is shortened a particular node's lifetime because it is frequently selected as a forward node, and it is affected the overall network lifetime. Moreover, although PDP algorithm selects the forward nodes as the optimized 1-hop neighbor nodes to cover 2-hop neighbor nodes, it can't make the optimized routing path in terms of overall sensor network.

The purpose of our proposed algorithm is to increase the sensor network lifetime. For the purpose, the algorithm selects the forward nodes taking account of node's

energy to disperse the energy dissipation on the sensor network and separates the selection process of forwarding node list into two phases to acquire the routing path to approximate the optimal routing path.

The nodes in sensor network exchanges information and maintains neighbor node table that send/receive 'Hello' and broadcast messages. Therefore, for proposed algorithm, we add node's energy information to neighbor node table, and add sender's energy information and uncovered 2-hop node set to broadcast message.

The basic concept of algorithm is as follows. First, a node to be send message selects the forward nodes using greedy set cover algorithm[5] among nodes with energy more than the average energy of 1-hop neighbor nodes. And it broadcasts the message. Since our algorithm selects the forward nodes only to nodes more than the average energy, it occurs the uncovered 2-hop neighbor nodes when the time elapsed. In this case, the broadcast message ( $UC_{node}$  field) includes the uncovered node list. Since a node in sensor network have several neighbor nodes, the uncovered nodes in current step can be covered by another neighbor nodes at the next step. When node receives the message, if node is the forward node, it executes the forward node selection process to select next forward nodes, and broadcasts the message; otherwise, it changes the sender's energy value in neighbor node table.

Since nodes in the sensor network have several neighbor nodes, it increases the redundant transmissions if forwarding step is increased. Using *CheckBit* of broadcast message, our algorithm makes the routing path in terms of overall sensor network. Therefore, it can reduce the redundant transmissions.

The detailed algorithm and execution process describes next subsection.

### 3.2 Algorithm Description

Figure 2 represents a pseudo code for the proposed algorithm to create the forwarding node list. The input of the proposed algorithm is  $U(u, v)$  and  $B(u, v)$  to be computed by PDP algorithm, *CheckBit*, and the uncovered 2-hop node set  $UC_{node}$  of the node  $u$ . The output is the forwarding node list  $F$  to select by node  $v$ , modified *CheckBit*, and the uncovered 2-hop node set  $UC_{node}$  of the node  $v$ . And  $Z$  denotes the a subset of  $U(u, v)$  covered so far,  $K$  denotes the set of  $S_i$ , and  $S_i$  denotes the neighbor set of  $v_i$  in  $U(u, v)$ . In this place,  $u$  is sender and  $v$  is receiver. Table 1 summarizes these terminologies.

The selection process of forwarding node list separates into two phases using *CheckBit*. In the first phase (*CheckBit* = 0), it selects the forward nodes among nodes more than the average energy of 1-hop neighbor nodes. Set *CheckBit* = 1, and add the forward node set to packet. In the other phase (*CheckBit* = 1), it selects the forward node to one node that has the maximum number of uncovered neighbors in 2-hop neighbor nodes among nodes more than the average energy of 1-hop neighbor nodes. Set *CheckBit* = 0, and add a forward node to packet.

The step by step description is provided as follows.

- Step 1** (Line 01-02) Initialize  $F$ ,  $Z$ , and  $K$  to use in algorithm. Add the node set to be covered by received node and the node set that the sender is not covered
- Step 2** (Line 03-06) For  $v_i \in B(u, v)$ , if energy of node  $v_i$  is more than the average energy of  $N(v)$ , find the intersection  $S_i$  of  $N(v_i)$  and  $U(u, v)$ , and add to  $K$ .

```

Input:  $U(u, v), B(u, v), CheckBit, UC_{node}$ 
Output:  $F, CheckBit, UC_{node}$ 

01: Let  $F = []$  (empty list),  $Z = \emptyset$  (empty set),  $K = \emptyset$  (empty set)
02:  $U(u, v) \leftarrow U(u, v) \cup UC_{node}$ 
03: for ( $v_i \in B(u, v)$ ) do
04:   if ( $E(v_i) \geq Avg\_E(v)$ ) then
05:      $S_i \leftarrow N(v_i) \cap U(u, v); \quad K \leftarrow \cup S_i$ 
06:   endif
07: do /* do-while loop */
08:   Find set  $S_i$  whose size is maximum in  $K$ 
      /* In case of a tie, the one with the smallest ID  $i$  is selected. */
09:    $F \leftarrow F \parallel v_k$ 
10:    $Z \leftarrow Z \cup S_i$ 
11:    $UC_{node} \leftarrow UC_{node} - S_i$ 
12:    $K \leftarrow K - S_i$ 
13:   for ( $S_j \in K$ ) do  $S_j \leftarrow S_j - S_i$ 
14: while ( $(K \neq \emptyset)$  and ( $CheckBit = 0$ ))
15: if ( $CheckBit = 0$ ) then  $CheckBit \leftarrow 1$ ; else  $CheckBit \leftarrow 0$  endif
16: if ( $Z = U(u, v)$ ) then exit
17: if ( $UC_{node} = \text{null}$ ) then
18:    $UC_{node} \leftarrow U(u, v) - Z$ 
19: else
20:   Find all possible node to be covered  $UC_{node}$  among  $N(v)$  and add to  $F$ 
21:    $UC_{node} \leftarrow \text{remainder } UC_{node} \cup (U(u, v) - Z)$ 
22: endif

```

Fig. 2. Pseudo code of the proposed algorithm

Table 1. Notation

Notation	Meaning
$U(u, v)$	the node set to be covered by node $v$
$B(u, v)$	Potential forward node set to cover $U(u, v)$
$UC_{node}$	the node set in $N(N(u))$ to uncovered node by node $u$
$F$	forward node list
$Z$	a subset of $U(u, v)$ covered so far
$S_i$	the neighbor set of $v_i$ in $U(u, v)$
$K$	the set of $S_i$
$Avg\_E(v)$	Average energy of $N(v)$
$E(v_i)$	energy of node $v_i$

**Step 3** (Line 08-13) Find set  $S_i$  whose size is maximum in  $K$  (in case a tie, the one with smallest ID  $i$  is selected). Add node with the detected  $S_i$  to the forward node list  $F$ , and adds the detected  $S_i$  to a covered node set  $Z$ . Removes the



detected  $S_i$  from the uncovered 2-hop node set  $UC_{node}$  of the sender and  $K$ . Remove  $S_i$  from the remainder subsets of  $K$ .

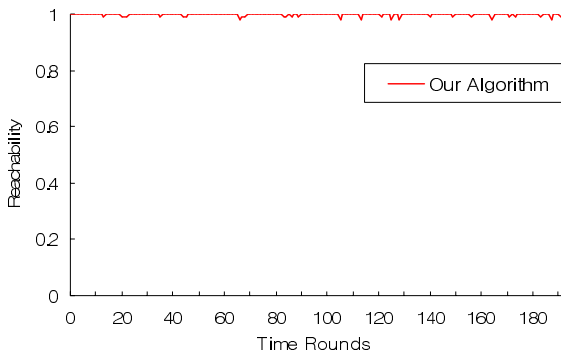
- Step 4** (Line 14) If  $K$  is empty or  $CheckBit$  is 1, it breaks do-while loop (i.e., do-while loop (line 07-14) executes once only and one node selects the forward node). Otherwise, repeat step 3 until  $K$  is empty.
- Step 5** (Line 15) Change the received  $CheckBit$  value.
- Step 6** (Line 16) If all 2-hop neighbor nodes is covered, the algorithm is finished.
- Step 7** (Line 17-18) If  $UC_{node}$  of the sender is null, add uncovered 2-hop neighbor node to  $UC_{node}$ .
- Step 8** (Line 19-22) Find 1-hop neighbor nodes whose can cover  $UC_{node}$  regardless of node's energy, add it to  $F$ , and remove covered node from  $UC_{node}$ . Add uncovered 2-hop neighbor nodes to  $UC_{node}$ .

## 4 Simulations

In this section, we evaluate the performance of the PDP algorithm and our algorithm in terms of the following evaluation bases: reachability (the number of all sensors receiving a packet), the number of forward nodes, the average number of packets a node receives, and the network lifetime (the number of alive sensor nodes).

The simulator randomly generates a connected unit disk graph within a broadcast area of  $m \times m$  (with  $m = 100$ ). Graphs are generated in two ways: a fixed transmitter range ( $r$ ) and a fixed average node degree ( $\bar{d}$ ). The average node degree is the expected number of nodes that are within a node's transmitter range. Specially, the average node degree can be approximated as  $\bar{d} = (\pi r^2 / m^2) \times n$  [2]. The number of hosts is 30, 60, 90 and 120. The node transmitter range is supposed 30, 40, 50 and 60. The simulation is conducted under the static environment. Assumed that dissipate the node's energy only when send or receive the messages (packets).

When a source node broadcasts a packet(broadcast message), each intermediate node will decide whether to retransmit the packet or to drop it independently, based on a given termination criterion. In other words, the broadcast process at each node will terminate when a given termination criterion is satisfied. In this paper, we suppose the following termination criteria. (Since each termination is decided locally, this approach corresponds to a reasonable termination criterion in a real system.)



**Fig. 3.** The reachability of proposed algorithm

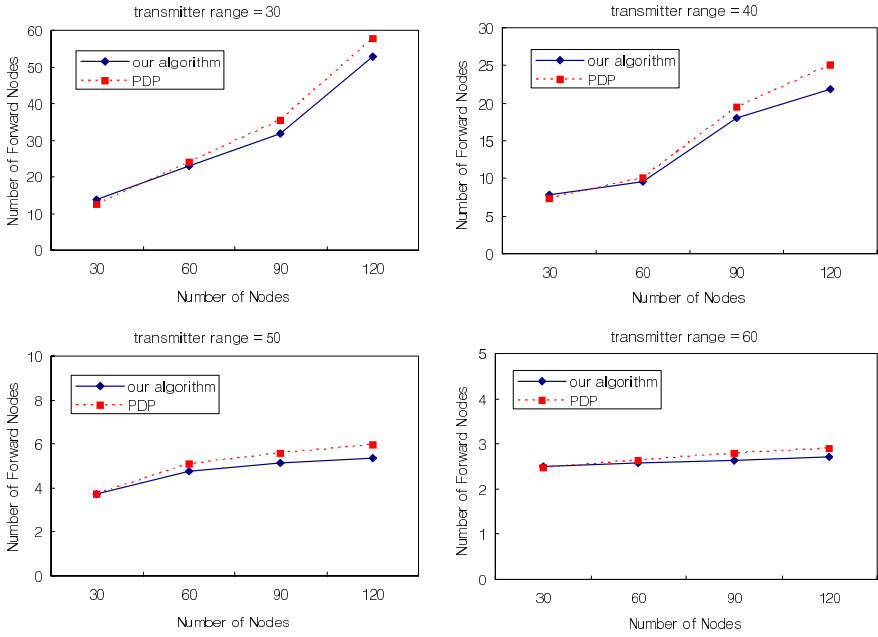


Fig. 4. The average number of forward nodes with the relayed/un-relayed criterion

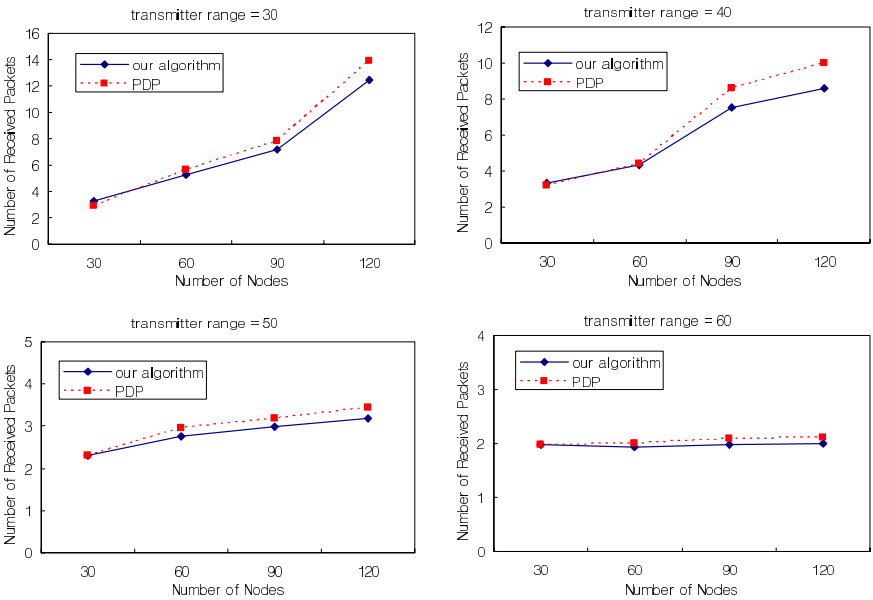
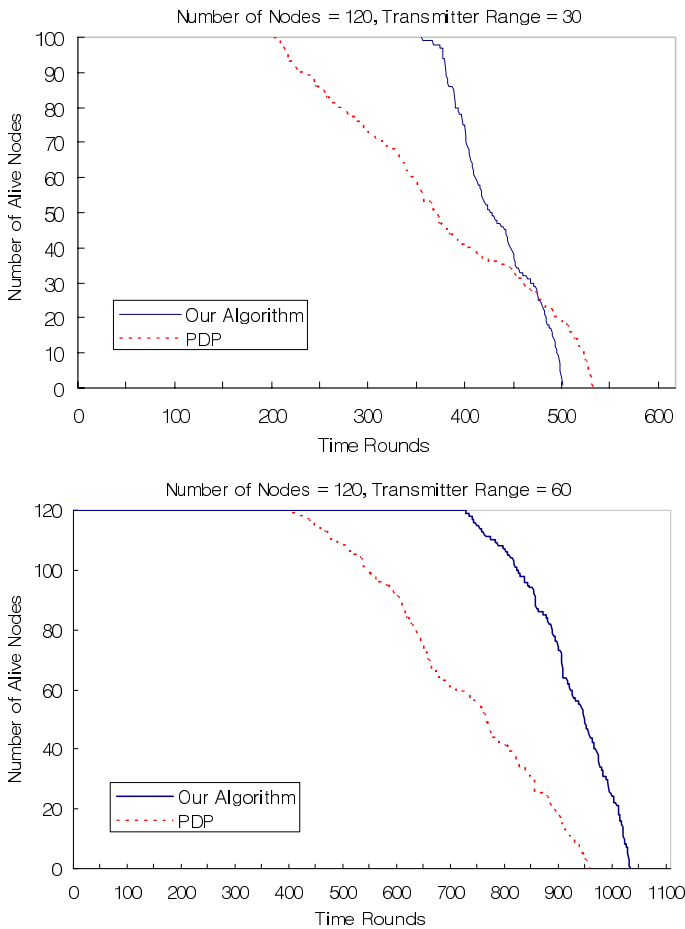


Fig. 5. The average number of packets a node receives with the relayed/un-relayed criterion

- The broadcast process (including the forward node selection and the broadcast process itself) is done quickly so that  $N(v)$  and  $N(N(v))$  remain the same during the process for each host  $v$ .
- Each node assigns a relayed/un-relayed status. A node  $v$  is called relayed when  $v$  has a packet; otherwise,  $v$  is called un-relayed.
- Forward node  $v$  will stop rebroadcasting a packet if  $v$  is relayed status.

Figure 3 shows the reachability of our algorithm. Figure 4 shows the average numbers of forward nodes and figure 5 shows the average numbers of packets a node receives during the broadcast process under different algorithms. Also, Figure 6 shows the network lifetime under different algorithms. (Figure 4 and 5 is the simulation results under all nodes alive.)



**Fig. 6.** Network lifetime (the number of alive nodes)

Figure 3 is the simulation results on reliability among performance evaluation criterion of broadcast protocol. The reliability represents reachability, i.e., the ratio of nodes connected to the source that received the broadcast message. The proposed algorithm records about 100% in terms of the message reception rate.

Figure 4 and 5 show the simulation results of the average number of forward nodes and the average number of broadcast packets that a node receives during the broadcast process for the fixed number of node (30, 60, 90 and 120), under relayed/un-relayed termination criteria. Our proposed algorithm shows the performance better than PDP algorithm when the number of nodes and transmitter range is increases. But, when the node density is low, our algorithm shows the performance same or less than PDP algorithm.

Figure 6 is the simulation results of network lifetime based on network transmitter range. As mentioned early (section 2.2), the lifetime of the sensor network can group into three classes by the kind of service it provides; when the first node dies, the half-life period of a sensor network, and the overall lifetime of a sensor network. In figure 6, it knows that the performance of our algorithm is better than PDP algorithm.

## 5 Conclusion

In this paper, we introduced a broadcast method that prolongs the network lifetime. For increase the sensor network lifetime, our proposed algorithm selects the forward nodes taking account of node's energy to disperse the energy dissipation on the sensor network and separates the selection process of forwarding node list into two phases to acquire the routing path to approximate the optimal routing path. Simulation results of applying this algorithm show that the proposed method in this paper has achieved better performance than the PDP algorithm in the lifetime of the network.

**Acknowledgments.** This paper was supported by special research funds of Seoul Woman's University in 2005.

## References

1. Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci, "A Survey on Sensor Networks", *IEEE Communications Magazine*, pp.102-114, Aug. 2002.
2. Wei Lou and Jie Wu, "On Reducing Broadcast Redundancy in Ad Hoc Wireless Networks", *Proc. HICSS'03*, pp.305b, Jan. 2003.
3. H. Lim and C. Kim, "Flooding in wireless ad hoc networks", *Computer Communications Journal*, 24(3-4):353-363, 2001.
4. Ivan Stojmenovic and Jie Wu, "Broadcasting and Activity-Scheduling in Ad Hoc Networks", *Mobile Ad Hoc Networking*, IEEE/Wiley, pp.205-229, 2004.
5. L. Lovasz, "On the ratio of optimal integral and fractional covers", *Discrete Mathematics*, vol. 13, pp. 383-390, 1975.
6. K. M. Alzoubi, P. J. Wan, and O. Frieder, "New distributed algorithm for connected dominating set in wireless ad hoc networks", *Proc. HICSS-35*, Jan. 2002.

7. G. Calinescu, I. Mandoiu, P. J. Wan, and A. Zelikovsky, "Selecting forwarding neighbors in wireless ad hoc networks", Proc. ACM DIALM'2001, pp.34-43, Dec. 2001.
8. A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint relaying for flooding broadcast message in mobile wireless networks", Proc. HICSS-35, Jan. 2002.
9. I. Stojmenovic, S. Seddigh, and J. Zunic, "Dominating sets and neighbor elimination based broadcasting algorithms in wireless networks", IEEE Trans. on Parallel and Distributed Systems, 13(1):14-25, Jan. 2002.
10. B. N. Clark, C. J. Colbourn, and D. S. Johnson, "Unit Disk Graph", Discrete Mathematics, vol. 86, pp. 165-177, 1990.

# Anonymous Routing in Wireless Mobile Ad Hoc Networks to Prevent Location Disclosure Attacks

Arjan Durresi<sup>1</sup>, Vamsi Paruchuri<sup>1</sup>, Mimoza Durresi<sup>2</sup>, and Leonard Barolli<sup>2</sup>

<sup>1</sup> Department of Computer Science, Louisiana State University,  
298 Coates Hall, Baton Rouge, LA, 70803, USA  
{durresi, paruchuri}@csc.lsu.edu  
<http://www.csc.lsu.edu/~durresi>

<sup>2</sup> Department of Information and Communication Engineering,  
Faculty of Information Engineering, Fukuoka Institute of Technology,  
3-30-1 Wajiro-Higashi, Higashi-ku, Fukuoka 811-0295, Japan  
durresim@franklin.edu, barolli@fit.ac.jp

**Abstract.** Wireless Ad Hoc networks are particularly vulnerable due to their fundamental characteristics such as an open medium, dynamic topology, distributed cooperation and constrained capability. Location information of nodes can be critical in wireless ad hoc networks, especially in those deployed for military purposes. In this paper, we present a set of protocols for anonymous routing to prevent location disclosure attacks in wireless ad hoc networks.

## 1 Introduction

Recent wireless research indicates that wireless Mobile Ad Hoc Networks (MANET) present larger security problems than conventional wired and wireless networks [1],[2]. In the traditional Internet, routers within the central parts of the network are owned by a few well-known operators and are therefore assumed to be somewhat trustworthy. This assumption no longer holds in an Ad Hoc network, since all nodes entering the network are expected to take part in routing. Also, because the links are usually wireless, any security that was gained because of the difficulty of tapping into the network is lost. Furthermore, because the topology in such a network can be highly dynamic, traditional routing protocols can no longer be used. Thus, Ad Hoc network has much harder security requirements than the traditional network and the routing in Ad Hoc networks is an especially hard task to accomplish securely, robustly and efficiently.

In general, wireless MANETs are particularly vulnerable due to their fundamental characteristics such as open medium, dynamic topology, absence of central authorities, distributed cooperation and constrained capability. The existing security solutions for wired networks cannot be applied directly in wireless MANETs.

Applications that make use of ad hoc routing have heterogeneous security requirements. Authentication, message integrity, and non-repudiation to an ad hoc environment are part of a minimal security policy. Apart from these, there

are several other security issues [1],[3] such as black hole attacks, denial of service, and information disclosure.

A location disclosure attack can reveal something about the locations of nodes or the structure of the network. The information gained might reveal which other nodes are adjacent to the target, or the physical location of a node. In the end, the attacker knows which nodes are situated on the route to the target node. If the locations of some of the intermediary nodes are known, one can gain information about the location of the target as well.

In many cases, the location information might be very crucial. In MANETs installed for tactical/military missions in a hostile and/or unknown territory, these types of attacks have to be prevented. In many cases, the communicating nodes need to be anonymous - no other node in the network should know who is communicating with whom. The organization of the paper is as follows: In Section 2 we explain the goals of our work, in Section 3 we summarize the background work, in Section 4 we present PAR, a solution that achieves complete anonymity and discuss trade-offs between complete anonymity and difficulty in identifying misbehaving nodes, in Section 5 we present enhancements to our protocol, in Section 6 we present our conclusions.

## 2 Goals

### 2.1 Anonymity

There are three types of anonymous communication properties: sender anonymity, receiver anonymity, and unlinkability of sender and receiver [16],[17]. Sender anonymity means that the identity of the party who sent a message is hidden, while its receiver (and the message itself) might not be. Receiver anonymity similarly means that the identity of the receiver is hidden. Unlinkability of sender and receiver means that although the sender and receiver can each be identified as participating in some communication, they cannot be identified as communicating with each other.

We also specify two degrees of anonymity - absolute and quasi-absolute. For simplicity, we describe these with respect to sender anonymity, but they can be extended to receiver anonymity and unlinkability as well. With absolute privacy, the attacker can in no way distinguish the situations in which a potential sender/receiver actually sent/received communication and those in which it did not. That is, sending a message results in no observable effects for the attacker. The identity and the location of each node in the network remain anonymous and sender and receiver are unlinkable. In case of quasi-absolute privacy, the neighboring nodes of the sender would be able to identify it, when it instantiates a connection and each node in the path of an established connection knows its next and previous hop neighbors' identities. Nonetheless, the receiver remains anonymous and the sender and receiver remain unlinkable unless all the nodes in the path collaborate.

It might be impossible to conceal the location itself - upon receiving a packet from a neighbor, as a node can easily approximate the location of the neighbor

by measuring the received signal strength and angle of arrival [23]. But, here we focus on making it impossible for a node to make a correspondence between the locations and the identities of the nodes. Thus, even though a node is able to figure out that some node is present at a particular location, it would not be able to find out the identity of that node.

## 2.2 What we Achieve

Initially, we present Protocol for Anonymous Routing (PAR), based on public key cryptography, to provide absolute anonymity. With this we achieve complete sender and receiver anonymity. Also, the sender and the receiver cannot be linked to each other even if all the nodes in the established path collaborate. However, with absolute anonymity, defending against denial-of-service attacks by compromised nodes becomes very difficult. We discuss this issue in more detail in Section 4. Hence, to detect and defend against these attacks, we present PAR-Enhanced, a variation of the above protocol, which only provides quasi-absolute anonymity as discussed in Section 5.

We consider the anonymity properties provided to an individual node against two distinct types of attackers:

- A local eavesdropper is an attacker who is also the neighbor of the sender/receiver and hence, can observe all (and only) communication to and from the sender/receiver.
- Collaborating nodes are other nodes that can pool their information.

PAR guaranties sender and receiver anonymity as well as unlinkability between sender and receive against both local eavesdroppers and collaborating attackers. Compared to PAR, in PAR-E the sender or the receiver are exposed to local eavesdroppers.

## 2.3 What we do not Achieve

Our objective is to provide anonymity for the sender and the receiver. In MANETs deployed specifically for military and tactical reasons, the identity and location information of the sender and the receiver might be critical. With absolute anonymity, the identities of every node in the network are completely anonymous. But, absolute anonymity makes it difficult, if not impossible, to detect misbehaving and compromised nodes in the network. A malicious node can refuse to forward packets or may just inject unnecessary packets into the network thus resulting in denial-of-service. Even intrusion detection systems [13],[14],[15] will be of little use. For networks where absolute anonymity is not as critical as in military networks, we can trade absolute anonymity a little so as to make detection of misbehaving nodes easy.

The protocol PAR that we present for achieving absolute anonymity makes no effort to defend against denial-of-service attacks. We believe that such attacks are inherent to networks where nodes are completely anonymous. But, with PAR-E, which provides quasi-absolute anonymity, intrusion detection systems [13], [14], [15] can be used with little or no modifications to detect misbehaving nodes.



### 3 Related Work

Ad hoc wireless networks assume no pre-deployed infrastructure for routing packets end-to-end in a network, and instead rely on intermediary peers. Securing ad hoc routing presents challenges because each user brings to the network their own mobile unit, without the centralized policy or control of a traditional network. Many ad hoc routing protocols such as Dynamic Source Routing (DSR), Ad Hoc On Demand Distance Vector (AODV), Zone Routing Protocol (ZRP), and Location Aided Routing (LAR) have been proposed previously, but none of the proposals have defined security requirements, and all inherently trust all participants.

All proposed protocols have security vulnerabilities and exposures that easily allow for routing attacks. These vulnerabilities are common to many protocols. The fundamental differences between ad hoc networks and standard IP networks necessitate the development of new security services. In particular, the measures proposed for IPSec [7] help only in end-to-end authentication and security between two network entities that already have routing between them; IPSec does not secure the routing protocol. While mechanisms similar to those used in IPSec can be adapted to secure the routing, IPSec alone does not suffice.

This point has been recognized, and others have started to examine security problems in ad hoc networks. A solution that uses threshold cryptography as a mechanism for providing security to the network is presented in [8]. A method that ensures equal participation among members of the ad hoc group, and that gives each node the authority to issue certificates is presented in [9]. An effort to secure an existing ad hoc routing protocol has also recently been made available [10].

Apart from the above protocols, which try to deal with minimal security requirements like Authentication, message integrity, and non-repudiation, several other protocols were presented to deal with specific security issues encountered in MANETs. [4] presents the resurrecting duckling security policy model, which describes secure transient association of a device with multiple serialized owners. [5] presents a solution to prevent black hole attacks, [6] presents strategies for intrusion detection.

Anonymous communication for wired networks is a well-studied aspect. The concept of a mix is introduced in [18]. A single processor in the network, called a mix, serves as a relay. Each processor  $P$  that wants to send a message  $m$  to a processor  $Q$  encrypts  $m$  using  $Q$ 's public key to obtain  $m'$ . Then  $P$  encrypts the pair  $(m', q)$  using the public key of the mix. The mix decrypts the message and forward  $m'$  to  $q$ . This scheme has been extended in [19],[20], [21] where several mixes are used to cope with the possibility of compromising the single mix. Another approach is to interpose an additional party (an anonymizer [22] between the sender and receiver to hide sender's identity from the receiver. Both the approaches are not viable in an ad hoc network for several reasons. First, they are based on the assumption that the information of mixes is known a priori and hence, the sender can select the mixes appropriate to the receiver. This assumption is impractical in an ad hoc network. Second, the mixes/anonymizers are entrusted with

more responsibility and they can become single points of attack. Third, forwarding a packet through mixes/anonymizers results in much longer paths than the shortest paths possible, thus resulting in inefficient utilization of resources.

We address one routing attack that could easily happen in MANETs, the information disclosure problem. Specifically, we deal with the attack in which a malicious node may leak location information of other nodes.

## 4 Protocol for Anonymous Routing

In this section, we present a protocol to achieve absolute anonymity - the identities of the source and destinations are not known to any other node; after a connection is established, a node involved in the path does not even know its adjacent nodes in the path. Instead of containing source and destination information, packets moving along an anonymous connection contain only obscure information about next hop and previous hop.

### 4.1 Notation and Definitions

*Public and Private Keys:* We assume the presence of a Public Key Infrastructure. We denote the private and public keys of a node  $i$  as  $E_i$  and  $D_i$ . With  $E(M, k)$  and  $D(M, k)$  we denote the encryption and decryption of message  $M$  with key  $k$ .

A Hash function,  $H$ , is assumed to be used globally; i.e., every node is aware of  $H$  and uses  $H$  to get the hash values.

*Invisible Address ( $IA_i$ ):* We also define the invisible address ( $IA_i$ ) of a node  $i$  for a packet with a flow identifier  $FID$  as constructed by encrypting the address along with  $FID$  first with the private key and then, with the public key of  $i$ :  $IA_i = E(E((i, FID, timestamp, RP), E_i), D_i)$

$RP$  is the redundancy predicate. Node  $N$  to have its invisible address get verified, just presents  $m = E((i, FID, timestamp, RIP), E_i)$  to the verifier. For the message to be verified successfully the unencrypted message  $D(m, D_i)$  must fulfill the redundancy predicate and  $E(m, D_i)$  must be same as  $IA_i$ .

*Routing Flow Table:* Each node maintains a routing flow table (RFT), through which the node is able to forward a packet to the next node in the path. The information stored in each entry of the table is:

- Flow Identifier ( $FID$ ) set to the unique request identifier present in the route request.
- Invisible Previous node Address ( $IPA$ ) set to the invisible address of the node from which the route request is received.
- Invisible Next node Address ( $INA$ ) set to the invisible address of the node from which the route received is received (if at all received).
- Timer ( $T$ ) initialized upon the reception of a non-duplicate route request to  $Th$ . The time  $Th$  depends on the diameter of the network and could be set to the maximum Round Trip Time that could be possible in the network between any two nodes. The entry is deleted if a route reply is not received before the timer expires.

We also assume that the network is very loosely synchronized. This assumption is just to prevent replay attacks.

## 4.2 Route Requests

Whenever a node  $S$  wishes to communicate with a node  $D$ , it initiates the route discovery process. Route discovery allows any node in the ad hoc network to dynamically discover a route to any other node in the ad hoc network, whether directly reachable within wireless transmission range or reachable through one or more intermediate network hops through other nodes. A node initiating a route discovery broadcasts a route request, which may be received by those nodes within wireless transmission range of it.

The route request has the following fields:

- $FID$  (Unique request identifier, also referred to as unique flow identifier) is set by the source by encrypting its address ( $S$ ), destination address ( $D$ ) and a locally maintained sequence number ( $SEQ$ ) with the public key of  $S$ . This is used to detect duplicate route requests received at an intermediate node:  $FID = E((S, D, SEQ), DS)$
- $ESA$  (Encrypted Source Address) is constructed by encrypting source address, hash of  $FID$ , timestamp and the Redundancy Predicate ( $RP$ ) with the destination's public key. The hash of  $FID$  and the timestamp are to prevent replay attacks.  $ESA = E((S, H(FID), timestamp, RP), DD)$
- $EDA$  (Encrypted Destination Address) is constructed by encrypting destination address, hash of  $FID$ , timestamp and the Redundancy Predicate ( $RP$ ) with destination public key.  $EDA = E((D, H(FID), timestamp, RP), DD)$
- $ITA$  (Invisible Transmitter Address) is the invisible address of the node  $i$  transmitting the route request.  $ITA = E(E((i, FID, timestamp, RP), E_i), D_i)$

Whenever a node  $i$  that is not the destination receives a non-duplicate route request packet, it performs the following operations:

1. A new entry is added to the routing flow table with  $FID$  and  $IPA$  fields set to  $FID$  and  $ITA$  values of the route request packet.
2. The node checks if the route request is intended for it by decrypting the  $EDA$  with its private key  $E_i$  and if it is the case it proceeds to send a route reply (described below) and steps 3 and 4 are not executed.
3. The timer is initiated.
4. Invisible address is computed for the packet and the route request is retransmitted with its  $ITA$  set to the invisible address computed.

## 4.3 Route Replies

The destination after receiving the route request also adds a new entry to its  $RFT$  in a similar manner as above. The destination also validates the source by decrypting  $ESA$  with  $E_i$ . Then, the destination in order to establish a connection, constructs a route reply packet with the following fields:

- *FID* is set to the *FID* of the route request.
- *ESA* (Encrypted Source Address) is constructed by encrypting *D* (destination of route request), hash of *FID*, timestamp and the Redundancy Predicate (RP) with the source's public key. The hash of *FID* and the timestamp are to prevent replay attacks.  $ESA = E((D, H(FID), timestamp, RP), D_S)$
- *ITA* (Invisible Transmitter Address) is the invisible address of the node *i* transmitting the route request.  $ITA = E(E((i, FID, timestamp, RP), E_i), D_i)$
- *IFA* (Invisible Forwarder Address) is initially set to the *ITA* of the corresponding route request packet.

Whenever a node *i* that is not the source, receives a route reply packet, it performs the following operations:

1. An entry corresponding to *FID* is searched for in the *RFT*. If no entry is found, the packet is dropped and all further steps are skipped.
2. The *IFA* value is verified by checking for *RP*, its address, *FID* and the timestamp in  $D(D(IFA, D_i), E_i)$ . If the verification fails, the packet is dropped and all further steps are skipped.
3. The *INA* value of the entry corresponding to *FID* in *RFT* is set to *ITA* of the route reply and the timer of the corresponding entry is nullified.
4. The *INA* value of the route reply packet is set to the *ITA* value of the entry corresponding to *FID* and *ITA* value of the route reply is set to the invisible address of *i*. The route reply is then forwarded.

When the source receives the route reply, it can verify the destination address by decrypting the *ESA* and *EDA* fields in the route reply with its private key. After the verification, the source and destination can securely communicate with each other.

It should be noted that, no node in the network could make out the source or the destination of any packet/connection. Also, each node in the network does not even know the address of its neighboring node to which it is forwarding the packet. Thus, communication that is completely anonymous can be achieved. Also, apart from the overhead imposed due to the implementation of public key infrastructure, no extra overhead is imposed by our protocol. It should also be noted that, for each new connection, the route request is flooded over the whole network. To reduce the overhead, instead of pure flooding, protocols such as distance based flooding [11], gossip based flooding [12] or BSP [24] can be used.

## 5 Enhanced Protocol for Anonymous Routing (PAR-E)

With PAR, a malicious node can misuse the complete anonymity gained by transmitting fake routing requests. A misbehaving node, which drops the packets instead of retransmitting packets, can also go undetected. It is always a trade off between privacy and security. We propose a few enhancements to detect malicious and misbehaving nodes, albeit at the cost of complete anonymity.

With the enhancements, a node will know the identity of any of its neighbors only if those two nodes lie on the same path of some connection. For instance, consider two neighboring nodes A and B. A will know the identity of B only if A and B lie in the path of some connection. If no such connection exists, then A does not know B and vice versa.

We assume that all the nodes are aware of some symmetric key encryption algorithm and all nodes use the same symmetric key encryption algorithm. We denote the symmetric encryption and decryption processes of a message  $M$  with key  $k$  as  $E_s(M, k)$  and  $D_s(M, k)$ .

## 5.1 Enhancements

Routing flow table: Five new fields are added to each entry of  $RFT$ : 1)  $n$ , a large prime chosen by the source. 2)  $g$ , a primitive mod  $n$ . 3)  $x$ , a large integer chosen for each entry by the node maintaining the  $RFT$ . 4)  $PPK$  (Previous node Partial Key) set to the partial key of the node from which the route request is received. 5)  $NPK$  (Next node Partial Key) set to the partial key of the node from which the route reply is received (if at all received).

*Route request*: Three new fields are added to the route request packet: 1)  $n$ , a large prime chosen by the source. 2)  $g$ , a primitive mod  $n$ . 3)  $TPK$ , Transmitter partial key, computed and set by the transmitter as  $TPK = g^x \text{ mod } n$

*Route reply*: Five new fields are added to the route reply packet 1)  $n$ , large prime chosen by the source of route request. 2)  $g$ , a primitive mod  $n$  and chosen by  $S$ . 3)  $NPK$ , next node partial key set by the transmitter. 4)  $TV$ , Transmitter Verifier, set to the cipher text obtained by encrypting the transmitter's address and its signature with  $SK$  as key.  $SK$  is computed using  $PPK$  and  $x$  as  $SK = (PPK^x \text{ mod } n)$ , and  $TV = Es((Tr\_address, signature), SK)$  5)  $TV'$ , Previous Transmitter Verifier, set to the cipher text obtained by encrypting the transmitter's address and its signature with  $SK'$  as key.  $SK'$  is computed using  $NPK$  and  $x$  as  $SK' = (NPK^x \text{ mod } n)$  and  $TV' = Es((Tr\_address, signature), SK')$

Initially, the source node  $S$  chooses a large prime,  $n$  and  $g$ , such that  $g$  is primitive mod  $n$  and initializes the corresponding fields in the route request to these. Any other node, that is not the destination, upon reception of a route request, apart from steps 1 - 4 (Section 4.2), performs an additional step 3a: The node chooses a large integer  $x$  and sets the field  $x$  in the newly created entry to that integer. Set the  $PPK$  field in the  $RFT$  entry to  $TPK$  of the route request and reset the  $TPK$  entry of route request to  $g^x \text{ mod } n$ .

The destination  $D$  upon receiving the route request creates a new entry in its  $RFT$  and sets its fields to the corresponding fields of route request. It then generates a large integer  $x$  and computes the shared key according to Diffie-Hellman key Exchange algorithm as :  $SK = TPK^x \text{ mod } n$ .

Then,  $D$  constructs the route reply as explained in Section 4.3 with the new fields set in the following way:  $n$ , set to the large prime present in the route request.  $g$ , set to  $g$  present in the route request.  $NPK$ , next node partial key, set to  $(g^x \text{ mod } n)$ ,  $x$  being a large integer, chosen by  $D$ .  $SK$ , shared key, is calculated as  $SK = (TPK^x \text{ mod } n)$ ,  $TPK$  being transmitter partial key obtained from

the route request.  $TV$ , Transmitter Verifier, set to the cipher text obtained by encrypting the transmitter's address and its signature with  $SK$  as key.  $TV = E_s((D, signature), SK)$ .  $TV'$ , Transmitter Verifier.

A node  $i$ , that is not the source, upon reception of a route reply, apart from steps 1 - 4 (Section 4.3), performs an additional step 3a: The node computes the shared keys,  $SK$  and  $SK'$  as  $(NPK^x \bmod n)$  and  $(PPK^x \bmod n)$ ,  $x$  being the value in the  $RFT$  entry corresponding to  $FID$ . Using  $SK$ ,  $TV$  is verified by decrypting it with the  $SK$ . Upon verification, it sets  $NPK$  field of route reply to  $(g^x \bmod n)$ ,  $x$  taken from  $RFT$  entry. Using  $TPK$  from the  $RFT$  entry, the node calculates  $SK = (TPK^x \bmod n)$  and resets the Transmitter Verifier in the route reply to  $TV = E_s((i, signature), SK)$ . It then retransmits the route reply.

Also, after retransmitting route reply, then node  $i$  overhears the route reply its neighbor retransmits for  $TV'$  and verifies the signature. In case of a node next to source, the source explicitly transmits  $TV'$  for the node to verify its identity. When the source receives the route reply, it verifies the destination address by decrypting the  $ESA$  and  $EDA$  fields in the route reply with its private key. Thus, secure communication channel between the source and the destination is established. It should be observed that each node in the path established knows nothing more than the identities of its neighboring nodes in the path established. The identities of even other neighboring nodes are revealed. As each node knows the identity of its neighboring nodes in the paths established, Intrusion Detection Systems such as [13],[14],[15] can be implemented successfully to detect malicious and misbehaving routers.

## 6 Conclusion

In this paper we presented protocols for achieving anonymous routing in mobile ad hoc networks and thus, prevent location disclosure attacks. The protocol for Anonymous Routing (PAR) guarantees absolute anonymity, which itself might cause problems as it would become hard to identify malicious and misbehaving nodes. PAR-Enhanced trades off some anonymity to enable detection of malicious and misbehaving nodes.

## References

1. Vesa Karpijoki: Security in Ad hoc Networks. In Proceedings of the Helsinki University of Technology, Seminars on Network Security, Helsinki, Finland, 2000.
2. L. Zhou and Z. J. Haas: Securing ad hoc networks. IEEE Network Magazine, 13(6):24-30, November/December 1999.
3. Janne Lundberg: Routing Security in Ad Hoc Networks. <http://citeseer.nj.nec.com/400961.html>
4. F. Stajano and R.J. Anderson: The Resurrecting Duckling: Security Issues in Ad-Hoc Wireless Networks. Proc. Seventh Security Protocols Workshop, Lecture Notes in Computer Science 1796, Springer-Verlag, Berlin, 2000, pp. 172-182.
5. H. Deng, W. Li, D. Agrawal: Routing Security in Wireless Ad Hoc Networks. IEEE Communications Magazine, Oct. 2002, pp. 70-75.

6. Lakshmi Venkatraman and Dharma P. Agrawal: Strategies for Enhancing Routing Security in Protocols for Mobile Ad hoc Networks. JPDC Special Issue on Mobile Ad Hoc Networking and Computing, accepted for publication.
7. C. R. Davis: IPSec: Securing VPNs. McGraw-Hill, New York, NY, USA, 2000.
8. L. Zhou and J. Haas: Securing ad hoc networks. IEEE Network, 13(6): 24-30, 1999.
9. J. P. HuBaux, L. Buttyan, and S. Capkun: The quest for security in mobile ad hoc networks. In Proc. ACM MobiHoc, October 2001.
10. S. Yi, P. Naldurg, and R. Kravets: Security-aware ad hoc routing for wireless networks. Technical Report UIUCDCS-R-2001- 2241, UILU-ENG-2001-1748, University of Illinois at Urbana-Champaign, August 2001
11. S. Y. Ni et al: The Broadcast Storm Problem in a Mobile Ad Hoc Network. ACM MOBICOM, pp. 151-162, Aug' 1999.
12. Haas, Halpern, Li: Gossip based Ad Hoc Routing. In IEEE INFOCOM, June 2002.
13. Oleg Kachirski, Ratan Guha: Intrusion Detection Using Mobile Agents in Wireless Ad Hoc Networks, IEEE Workshop on Knowledge Media Networking.
14. R. Ramanujan, S. Kudige, T. Nguyen, S. Takkella, and F. Adelstein: Intrusion-Resistant Ad Hoc Wireless Networks, Proceedings of MILCOM 2002, Oct. 2002.
15. Yongguang Zhang and Wenke Lee: Intrusion Detection in Wireless Ad-Hoc Networks, Proceedings of The Sixth International Conference on Mobile Computing and Networking, Boston, MA, August 2000
16. Pfizmann, A. and Waidner, M: Networks without user observability. Computer Security, 6/2 (1987) 158-166.
17. M. K. Reiter and A. D. Rubin: Crowds: Anonymity for Web Transactions. ACM Transactions on Information and System Security, pp. 66-92, 1(1), 1998.
18. D. Chaum: Untraceable electronic mail, return addresses, and digital pseudonyms. Communication of the ACM, vol. 24, no. 2, pp. 84-88, 1981.
19. A. Pfizmann, B. Pfizmann, and M. Waidner: ISDN-MIXes - Untraceable communication with very small bandwidth overhead. In Proc. Kommunikation in verteilten Systemen, pp. 451-463, 1991.
20. C. Rackoff and D. Simon: Cryptographic defense against traffic analysis. In Proc. of the 25th Annu. ACM Symp. on the Theory of Computing, pp. 672- 681, 1993.
21. P. F. Syverson, D. M. Goldschlag, and M. G. Reed: Anonymous connections and onion routing. In Proc. of IEEE Symposium on Security and Privacy, pp. 44-54, 1997
22. Anonymizer - Online Privacy and Security. [www.anonymizer.com](http://www.anonymizer.com)
23. D. Niculescu and B. Nath: Ad Hoc Positioning System (APS) using AoA, INFOCOM. 03, San Francisco, CA, 2003.
24. Arjan Duresi, Vamsi Paruchuri, Sitharama Iyengar, Rajgopal Kannan: Optimized Broadcast Protocol for Sensor Networks, IEEE Transactions on Computers, Vol. 54, No. 8, August, 2005, pp. 1013-1024.

# The Design and Implementation of a Location-Aware Service Bundle Manager in Smart Space Environments\*

Minwoo Son, Soonyong Choi, Dongil Shin\*\*, and Dongkyoo Shin

Department of Computer Science and Engineering, Sejong University,  
98 Kunja-Dong, Kwangjin-Ku, Seoul 143-747, Korea  
{minwoo15, artjian, dshin, shindk}@gce.sejong.ac.kr

**Abstract.** Ubiquitous computing as the integration of sensors, smart devices, and intelligent technologies to form a “smart space” environment relies on the development of both middleware and networking technologies. Several kinds of smart space middleware have been developed and OSGi (Open Service Gateway Initiative) was initiated mainly for interoperability support with service distribution among various middleware environments. In this paper, we analyze the weaknesses in the OSGi service platform such as a non-distributed framework, passive user management, device management, and service bundle management. Moreover, we propose that in a smart space environment, Location-aware SBM (Service Bundle Manager) is most capable of efficiently managing various service bundles. This paper concludes with the implementation results for the SBM.

## 1 Introduction

Ubiquitous Computing means that users can use computers naturally and conveniently, regardless of place and time [1]. It means that a computer existing anywhere can use specialized services, and change its contents according to place or time via sensing and tracking. Its ability to form a “smart space” environment depends on the availability of networks, services, sensors, wireless communication and smart middleware technologies [2]. Smart space offers people security, energy saving, convenience, and a better lifestyle.

A smart space is a living and office environment in which devices can be accessed and controlled either locally or remotely. For example, a user may monitor the status of home appliances status from the office and switch them on/off as needed. When someone approaches the front door, smart middleware turns on the porch light. In short, smart services support people.

By connecting smart devices and intelligent networks, a smart space allows the user to access information efficiently and to connect directly to a range of public and personal services (including banks, police, fire, and emergency responders).

---

\* This study was supported by a grant of the Korea Health 21 R&D Project, Ministry of Health & Welfare, Republic of Korea. (0412-MI01-0416-0002).

\*\*Correspondence author.



Convenience and efficiency are maximized by controlling information-communication equipment, digital audio and video devices, other existing electronic devices, smart sensors, etc.

Middleware for a smart space controls home appliances, facilitates interaction among electronics, and supports various services. A variety of middleware for smart spaces has been developed, including UPnP (Universal Plug and Play) [3, 4], Jini [5, 6], HAVi (Home Audio Video Interoperability) [7, 8], IEEE 1394 [9, 10], and PLC (Power Line Communication) [11]. The shortcomings of smart space middleware are a lack of interoperability and difficulty of distributing new middleware-specific services. OSGi (Open Service Gateway Initiative) has been developed to overcome these problems by enabling the easy deployment of services for local smart spaces [12, 13].

OSGi is gradually extending its influence to the smart space middleware market, and electronic devices based on OSGi are being used. And to control home and office electronic devices based on OSGi, Service Bundles based on OSGi have been developed and also available. Therefore a user's need for an efficient manager has suddenly increased by several service bundles.

OSGi Spec. version 3 offers many services. For example it includes Framework for a service bundle manager and event processing, Log Service for logging information and retrieving current or previously recorded log information in the OSGi Service Platform, and Device Access Service for an electronic home appliance manager. However, the OSGi Service Platform does not support updating, installing, or removing for the active life-cycle of service bundles, and will not automatically check-in a device's state, or update a device driver, or distributed framework. Therefore we suggest SBM (Service Bundle Manager) to solve these problems.

This paper is composed of six sections. Section 2 and Section 3 introduce OSGi and describe a smart space architecture based on OSGi. In Section 4, we propose SBM to efficiently manage many kinds of service bundles based on OSGi and describe the related implementation results in Section 5. Finally we conclude in Section 6.

## 2 Background

Many kinds of projects are currently in progress with a shared perspective of exploring a new research agenda. Some of these projects are Easy Living [14], Smart-Its [15], SSLab [16], the Aware Home [17], and iRoom [18]. Microsoft's "Easy Living" project is developing prototype architectures and intelligent technologies which include multiple sensor modalities combine, automatic or semi-automatic sensor calibration and model building, and on the like for smart space environments.

Users use many smart devices that include each other's middleware in smart space. Therefore we implement smart space middleware with OSGi of smart space environment because OSGi supports communication among several pieces of middleware.

OSGi was created in 1999 in order to define an open standard for the delivery of services in networked environments, (vehicles and homes, for example.) and was supposed to solve problems involving the interaction among several kinds of home network middleware and service distribution. The OSGi service platform is an intermediary between the external network environment and the smart space network environment.

The OSGi service platform is divided into two parts: the OSGi Service Framework and OSGi Service. Figure 1 shows the OSGi Architecture. The OSGi framework supports service registry, life-cycle management of service bundles, etc. and includes registry service, persistent, data storage and life-cycle management for a service in an extendable Java runtime environment. The OSGi framework supports an execution environment for services. An OSGi service is defined by a Java Interface. OSGi services include HTTP, Logging, and Device Access Service. A service is implemented as part of a bundle. A bundle is the smallest unit of management for the framework and, the framework manages its installing, uninstalling, resolving, stopping, starting, and active life-cycle. A bundle consists of java class code files and additional resources. In addition, the framework has Manifest File which is Meta information for the bundle or a bundle’s install, start, and stop information. Several OSGi services may be included on a bundle, and form a standard unit of distribution and management.

Recently, research into the OSGi service platform suggests that a user in a smart space environment can turn appliances on and off. In other words, a smart space based on OSGi service platform supported a solid infrastructure so that projects could focus on unifying the smart space with smart phone and other smart applications [13].

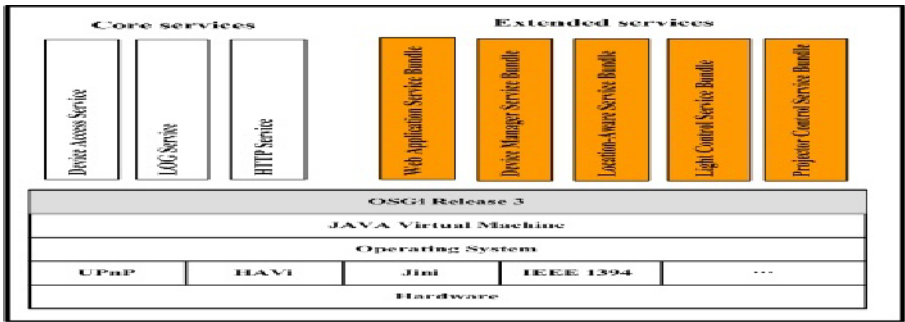
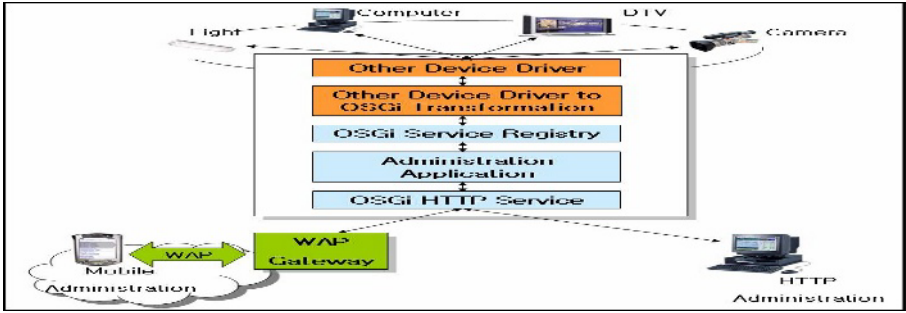


Fig. 1. OSGi Architecture

### 3 Smart Space Network Architecture Design Based on OSGi

OSGi, developed to resolve interoperability among pieces of equipment based on different middleware, may be activated in a smart space Network. Figure 3 is the design for a Smart Space Service Network Architecture to control devices that do not offer OSGi Device Drivers. When senses a new device, the Support Registry, the OSGi HTTP Service, is designed to support the control attachment of existing devices. For example, when DTV is a plug-in, it will confirm whether it is registered with the Service Registry and will support service or download drivers if necessary. To control devices based on other middleware (Jini, HAVi, UPnP, etc.), each device connects through service bundles that are driver modules based on other middleware and are implemented as Java packages. In Figure 2, the OSGi Transformation step checks that other device drivers for different middleware supports which kind of services. The service bundle is registered in the OSGi Service Registry step. Also, when other devices require communication, the bundle supports export services.

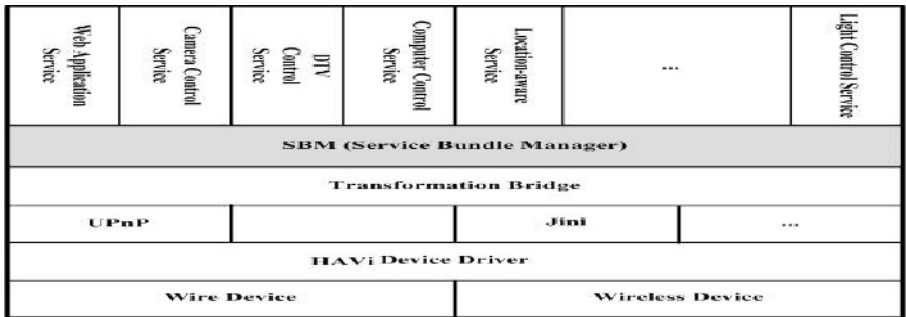


**Fig. 2.** Transform device based on other middleware into service based on OSGi in Smart Space Network Architecture

## 4 SBM (Service Bundle Manager) Design

### 4.1 Smart Space Gateway Architecture Design Based on SBM

Several service bundles will be used by a user, as a home network is widely used. Therefore users will need easier and more efficient manager service bundles. The OSGi service platform includes weaknesses for the management of service bundles. We compensated for the OSGi platform’s passive service element, user management, device manager component and non-distribution, etc. and designed SBM to efficiently manage several service bundles such as a Light Control Service and a Location-aware Service.



**Fig. 3.** Smart Space Gateway Architecture

Figure 3 shows the SBM-based Smart Space Gateway Architecture. The Device Driver and Wired/Wireless Device in the lower part decide the connection system among devices and certainly need standardization. Because the operating system uses programs like WinCE, embedded Linux, and real-time OS, it has less need of standardization. Connection systems for devices include wireless devices such as Wireless LAN, RFID (Radio Frequency Identification) [19, 20], Bluetooth [21, 22], and some of the wired devices consist of USB, IEEE 1394, and Ethernet. If a device

physically connects to a smart space network, it connects the new device to middleware such as UPnP, HAVi and Jini, which automatically reconstruct the smart space network.

Transformation Bridge supports communication between middleware. When OSGi decides on supportable middleware, the home gateway uses the appropriate Transformation Bridge.

Like the OS in a computer, Windows, Linux and Max decide on applications for the computer system, SBM based on OSGi, which is a home gateway in a smart space network, supports home network services, when SBM connects devices inside or outside of the smart space. It is used to control service bundles such as the Web Application Service, Camera Control Service, and the Device Manager Service. SBM solves weaknesses in the service platform for OSGi Spec. version 3, such as passive service, User Management, Device Management and non-distribution.

### 4.2 SBM Architecture Design

Figure 4 shows SBM architecture. To control home appliances, a user uses two connection systems.

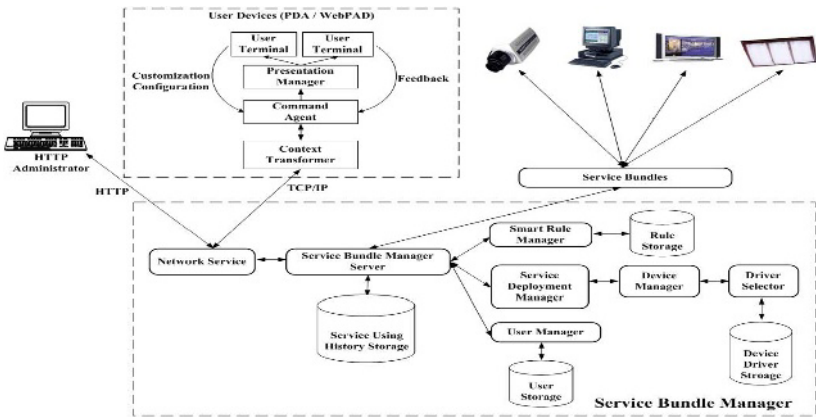


Fig. 4. Service Bundle Manager Architecture

The first method makes it possible for a user to control a service bundle in SBM, after the user is authenticated through a web browser.

The second method is a HIML (Human Interface Markup Language)[23] document that transmits using mobile devices such as PDA or Web PAD using a Network Service to Service Bundle Manager Server approach. A HIML document is stored to Service Using History Storage and the document is analyzed.

The HIML document pattern is made according to the data form of the HIML document to divide electronic devices, as seen in Figure 5, into image devices and sound devices; the DataSet is shown here.

To control electronic devices through using a web or mobile device, it accepts data on access privileges by a user's device in User Manager through a user ID if users

approach. After the Device Manager receives an electronic device ID and device function services, it finds an appropriate driver through the device. Finally, users can control devices by service bundles.

When each service bundle starts, the Service Bundle Manager Server checks the Smart Rule Manager. The Smart Rule Manager then checks Rule Storage, which includes start-rule lists of service bundles. If the service bundle’s start-rule exists, Smart Rule Manager sends Service Bundle Manager Server service bundle’s start-rule. The Service Bundle Manager Server controls the service bundle through the service bundle start-rule. Rule Storage includes theses that support auto-aware system and are managed by the user (rule list installs, remove, modify, etc.).

**Table 1.** Storage Implements

Storage	Implements
<b>Service Using History Storage</b>	<ul style="list-style-type: none"> <li>· stores HIML, which mobile device sends, to control device</li> <li>· makes each device table</li> <li>· connects HIML documents with Context Awareness data</li> </ul>
<b>User Storage</b>	<ul style="list-style-type: none"> <li>· stores users’ personal information such as id, name, age, career, etc.</li> <li>· according to user id, allows certified user to access device control.</li> <li>· supports fitting device services to recognized user.</li> </ul>
<b>Device Driver Storage</b>	<ul style="list-style-type: none"> <li>· saves driver of each device.</li> <li>· driver always uses the latest version.</li> </ul>
<b>Rule Storage</b>	<ul style="list-style-type: none"> <li>· includes intelligent rule.</li> <li>· rules are inserted, removed, modified by user and auto-modify.</li> </ul>

**Table 2.** Service Bundle Management Implements

Service	Implements
<b>Network Service</b>	<ul style="list-style-type: none"> <li>· connects SBM through Web, TCP/IP</li> <li>· supports service that controls service bundle.</li> </ul>
<b>Service Bundle Manager Server</b>	<ul style="list-style-type: none"> <li>· manages execution life-cycle of service bundle (install, start, stop, resume, uninstall).</li> <li>· user manages service bundle through network.</li> <li>· collects service bundle’s information</li> <li>· sends state information of service bundle to Administrator on schedule.</li> </ul>
<b>Service Deployment Manager</b>	<ul style="list-style-type: none"> <li>· links driver with fitting service bundle.</li> </ul>
<b>Device Manager</b>	<ul style="list-style-type: none"> <li>· manages each device’s driver and service (addition/insert/remove).</li> <li>· periodically updates driver and stores driver in driver storage.</li> </ul>
<b>Driver Selector</b>	<ul style="list-style-type: none"> <li>· selects best suited to a driver service which user wants.</li> </ul>
<b>User Manager</b>	<ul style="list-style-type: none"> <li>· manages user’s personal information (addition/insert/remove).</li> <li>· user level for limiting device control.</li> </ul>
<b>Smart Rule Manager</b>	<ul style="list-style-type: none"> <li>· checks service state of service bundles</li> <li>· sends Service Bundle Manager Server information of Rule Storage</li> </ul>

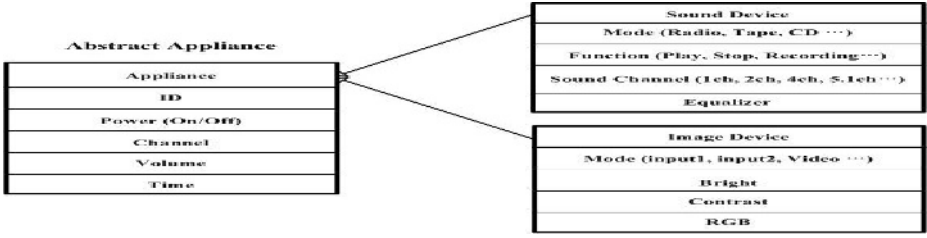


Fig. 5. DataSet

## 5 Implementation of Service Bundles Based on SBM

This paper utilized an OSGi Release 3 compliant environment.

We suggest a scenario to test SBM in smart space. After SBM recognizes a person’s location in the office through a Location-aware service bundle, and sends awareness-information to the Smart Rule Manager. Then the Service Bundle Manager Server turns light on and off through Light Control service bundle.

### 5.1 Light Control Service Bundle

Figure 6 shows the Light Control service bundle hierarchy. The Light Control service bundle turns the lights of homes and offices on and off. It can turn the light on and off automatically. SBM detects a user’s location through a Location-aware service bundle. The Smart Rule Manager supports several services according to the user’s location. For example, SBM automatically turns a light on or off from information provided by the Smart Rule Manager.

LightControl class controls which light goes on or off. After TimeZone class checks time, according to time (AM/PM), but only Light turn off at AM and turn on at PM automatically and at the same time Light Control Service Bundle is controlled light on/off by user.

SBM manages the transmission of data between Light Control Service and Client through the Client class to control the Light’s Channel during the Light Control Service Bundle’s run-time.

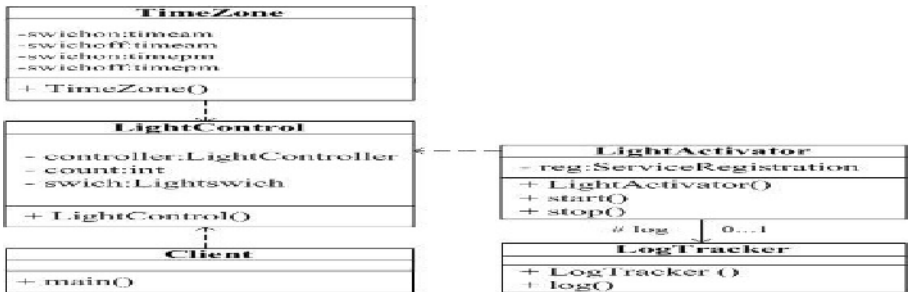


Fig. 6. Light Control Service Bundle hierarchy

### 5.2 Location-Aware Service Bundle

The Activator and Event Broker components approach the SBM framework, as shown in Figure 7. An Activator component, by interaction with the SBM framework, provides information on the Location-aware service bundle’s start and stop from the administration interface of the framework. The Related Client in the Location-aware service bundle Event records in Event Broker. If Location-aware service bundle receives an Event, Event Broker sends an event object to the client. According to this method, a Location-aware service bundle shares Camera Handler information and Position Recognition information with other service bundles (ex. Light Control service bundle) at the same time by multi application in the SBM framework.

Location-aware service views a user’s location in home and office in real-time through the application.

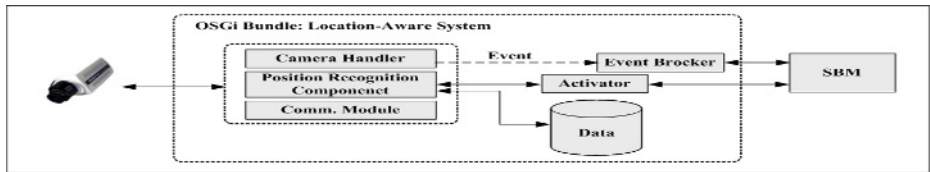


Fig. 7. Location-aware Service Bundle Architecture



Fig. 8. Location-aware Service Bundle Hierarchy

Figure 8 shows each class’ relation in Location-aware service bundle. Location\_awareActivator class controls a bundle’s states, such as start, install and stop. If calling start(), a service bundle scarcely starts when through CameraHandler, PositionRecognition and VisionProcessor classes perform. After CameraHandler class checks which camera attaches or detaches through Cam\_driver, the service bundle receives the user’s location information from PositionRecognition and VisionProcessor classes.

After Location-aware service bundle compare target picture, which is nobody in smart space environment, with real-time picture, the PositionRecognition and VisionProcessor class recognizes user’s location in smart space.

LogTracker Class processes the recording of events and errors. The log() method logs a message with an exception associated with a specific service. The class starts as soon as the service bundle starts and if the service bundle stops, the class calls the log.close() method to stop. Figure 9 shows recognition of the user's location in smart space environment during the Location-aware service bundle's run-time. Then SBM turns light on through Light Control service bundle on the user's location.



**Fig.9.** View Location-aware Service Bundle in Smart Space

## 6 Conclusion

This paper proposes SBM, which efficiently manages several service bundles based on OSGi and describes the execution of service bundles in an SBM environment.

SBM, which solves the OSGi service platform's weaknesses, such as User Management and Device Management, permits certified users to control each device and automatically designs a service for each device. After a user enters SBM using a web service and mobile device for the control of a device, SBM controls the sending of the device's service information, which analyses access privileges through User Manager and Device Manager, to the server. We did research on service bundles in a smart space system and on a manager for home appliances' control and user's location awareness service. SBM updates service bundles automatically and efficiently manages service bundles by managing a user's authorization and by controlling each device.

Future work will be done on a study of SBM to manage home appliances and service bundles, after extending its services such as context awareness, authenticated security and distribution. Smart Rule Manager parts and security still leave something to be desired.

## References

1. Schulzrinne. H., Xiaotao. Wu, Sidiroglou. S., Berger. S.: Ubiquitous computing in home networks, Communications Magazine, IEEE, Vol. 41, Issue. 11, (2003) 128 - 135
2. George Alyfantis, Stathes Hadjiefthymiades, Lazaros Merakos: A Smart Spaces System for Pervasive Computing, EDBT 2004 Workshops, Vol. 3268, (2004) 375-384
3. UPnP Specification v1.0 homepage, <http://www.upnp.org>



4. Dong-Sung Kim, Jae-Min Lee, Wook Hyun Kwon, In Kwan Yuh: Design and implementation of home network systems using UPnP middleware for networked appliances, *Consumer Electronics, IEEE Transactions on*, Vol. 48, Issue. 4, (2002) 963 - 972
5. Jini Specification v1.0 homepage, <http://www.jini.org>
6. Landis. S., Vasudevan. V.: Reaching out to the cell phone with Jini, *System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on*, (2002) 3821-3830
7. HAVi Specification v1.1 homepage, <http://www.havi.org>
8. Lea. R., Gibbs. S., Dara-Abrams. A., Eytchison. E.: Networking home entertainment devices with HAVi, *Computer*, Vol. 33, Issue. 9, (2000) 35-43
9. IEEE 1394 Specification v1.0 homepage, <http://www.1394ta.org>
10. Nakagawa. M., Honggang Zhang, Sato. H.: Ubiquitous homelinks based on IEEE 1394 and ultra wideband solutions, *Communications Magazine, IEEE*, Vol. 41, Issue. 4, (2003) 74 - 82
11. Ferreira. H.C., Grove. H.M., Hooijen. O., Han Vinck, A.J.: Power line communications: an overview, *AFRICON, IEEE AFRICON 4th*, Vol. 2, (1996) 558 - 563
12. OSGi Specification v. 3.0, March.2003 homepage, <http://www.osgi.org>
13. Choonhwa Lee, Nordstedt, D., Helal, S.: Enabling smart spaces with OSGi, *Pervasive Computing, IEEE*, Vol 2, Issue 3, (2003) 89-94
14. Berners-Lee, T., *Answers of Young People*, <http://www.w3.org/People/Berners-Lee/Kids>
15. The Aware Home, <http://www.cc.gatec.edu/fce/ahri>
16. Brumitt B., Meyers, B., Krumm, J., Kern, A. and Winograd T.: The interactive workspaces project: experiences with ubiquitous computing rooms", *IEEE Pervasive Computing*, Vol. 1, Issue. 2, (2002) 67-74
17. The Smart-Its Project, <http://www.smart-its.org/>
18. Smart Space Laboratory, <http://www.ht.sfc.keio.ac.jp/SSLab>
19. Radio Frequency Identification (RFID) homepage, <http://www.aimglobal.org/technologies/rfid>
20. Want. R.: Enabling ubiquitous sensing with RFID, *Computer*, Vol. 37, Issue. 4, (2004) 84-86
21. Bluetooth homepage, <http://www.bluetooth.com>
22. Kwang Yeol Lee, Jea Weon Choi: Remote-controlled home automation system via Bluetooth home network, *SICE 2003 Annual Conference*, Vol. 3, (2003) 2824-2829
23. Gunhee Kim, Dongkyoo Shin, Dongil Shin: Design of a Middleware and HIML(Human Interface Markup Language) for Context Aware Services in a Ubiquitous Computing Environment, *EUC 2004*, (2004) 682-691

# A Context-Aware and Augmented Reality-Supported Service Framework in Ubiquitous Environments

Jae Yeol Lee\* and Dong Woo Seo

Department of Industrial Engineering, Chonnam National University,  
300 Yongbong-dong, Buk-gu, Gwangju 500-757, South Korea  
jaeyeol@chonnam.ac.kr  
dongwooseo@paran.com

**Abstract.** We present a Context-Aware and Augmented Reality-supported service Framework for distributed and collaborative interactions in Ubiquitous environments (U-CAFÉ). The proposed approach adopts semantic web-based context management and reasoning for supporting context-aware services. Further, it utilizes augmented reality for providing more relevant and human-centered interactions and collaborations. A semantic web representation language, Topic Map, is used to manage and reason about ubiquitous service-related contexts explicitly and systematically. Augmented reality-based interactions are used for embedding virtual models onto physical models considering contexts and for enabling bi-augmentation between virtual and physical models. The proposed framework has been successfully applied to design collaboration and intelligent home services.

## 1 Introduction

Computing paradigm is moving toward context-aware and ubiquitous environments in which devices, software agents, and engineering services are all expected to seamlessly integrate and cooperate in support of human objectives – anticipating needs, negotiating for services, acting on our behalf, and delivering services in anywhere, any-time fashion [4,5,14]. Context-aware and ubiquitous systems are computer systems that can provide relevant services and information to users by exploiting contexts. By context, we mean information about locations, software agents, engineering services, users, devices, and their relationships. Contexts may also include system capabilities, services offered and sought, the activities and tasks in which people and computing entities are engaged, and their situational roles, beliefs, and intentions. Note that an effective semantic management for contexts is one of the key requirements for building a context-aware ubiquitous service framework. The Web Ontology Language [11] and Topic Map [16] are languages for expressing sophisticated class definitions and properties.

Augmented reality (AR) can naturally complement ubiquitous computing by providing an intuitive and collaborative interface to a three-dimensional information space embedded within physical reality [2]. Correspondingly, the human-computer

---

\* Corresponding author.

interfaces and interaction metaphors originating from AR research have proven advantageous in a variety of real-world ubiquitous application scenarios, such as industrial assembly and maintenance, location-based intelligent systems, navigation aides, and computer-supported cooperative work [7,9,12,13].

Although context-aware computing is very popular in the areas of building intelligent meeting rooms, supporting intelligent robots, and providing smart spaces for easy living [4,5,8,14], a more sophisticated research is still needed that combines context-aware computing with more natural and intuitive interfaces like augmented reality for supporting human-centered collaborative interactions which are indispensable to product design, distributed virtual reality, and human-computer interactions.

In this paper, we present a Context-aware and Augmented reality-supported service Framework for distributed and collaborative interactions in Ubiquitous environments (U-CAFÉ). The proposed approach adopts a semantic web-based context management of ubiquitous services. Topic Maps are used to query and reason about service-related contexts, which can reduce difficulty and cost in building context-related knowledge management and sharing that can provide more relevant services and information to meet service requestors on the basis of their contexts. Contexts for ubiquitous services are maintained in three different levels of details adoptable to a dynamically changing environment: 1) proxy generation for each context, 2) cluster generation based on the hierarchical representation of contexts among persons, devices, and service proxies, and 3) Topic Map-based integrated context map generation. Further, augmented reality is used to realize more human-centric interfaces and collaborations in which three-dimensional computer graphics are superimposed over real objects considering contexts. Collaborative interactions based on AR not only feedback to existing contexts or generate new contexts, but also get interactions from the contexts, which realizes b-augmentation between physical and virtual services. We also discuss how Web services and JINI services are used to register, lookup, and bind ubiquitous services easily and effectively [6,15]. The remainder of the paper is organized as follows. Section 2 overviews U-CAFÉ. Section 3 presents how to maintain service contexts and apply them to augmented reality in a ubiquitous environment. Section 4 shows some implementation results. Finally, section 5 concludes with some remarks.

## 2 System Overview

The primary objective of this research is to propose a generic framework that supports collaborative and adaptive capabilities in a ubiquitous and context-aware environment as shown in Fig. 1. The framework has been built on the three layers: 1) U-service layer, 2) U-context layer, and AR-based collaboration layer. The U-service layer works as a service dispatching and aggregation broker. It supports dynamic ubiquitous service federations via process templates. Readers are referred to see the Reference 10 for detailed description of process-centric service federations. Published Web services communicate directly with their legacy applications by Web service wrapping [15]. The U-context layer maintains contexts from various resources such as devices, people, environment, etc. Further, the U-context broker facilitates reasoning and querying of contexts represented in Topic Map. Based on these contexts, requestors or mobile devices can dynamically adapt to the most desirable situation to analyze the requested services. The AR-based collaboration layer provides more realistic and

human-oriented services. It is linked to the U-service and U-context layers for context acquisition and reasoning, and graphical information gathering and synchronization. Thus, the three layered framework can support various kinds of ubiquitous services and collaborations such as context-aware adaptation to the environment and human-centered AR-based collaborations.

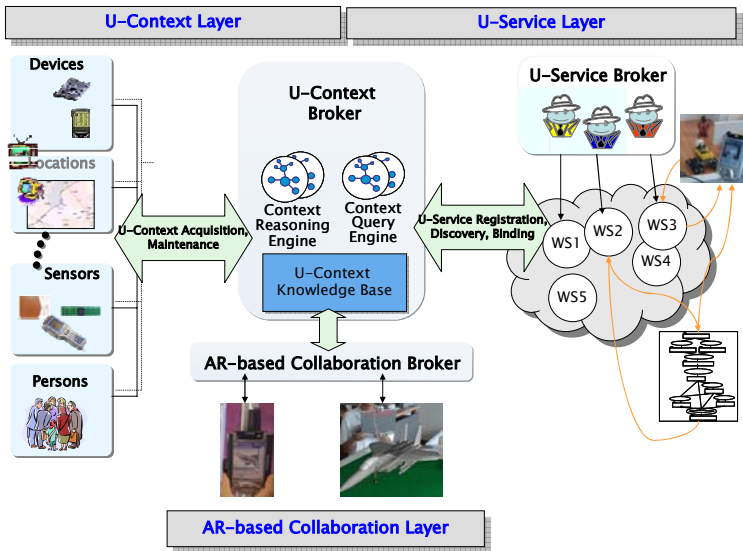


Fig. 1. Conceptual implementation of U-CAFÉ

Both Web services and JINI™ services are used for flexible and easy communication to support various kinds of engineering services and devices. Each ubiquitous service can be easily represented as a Web service component. Further, service federation can be achieved by utilizing BPEL4WS (Business Process Execution Language for Web Services), which can generate another new service [3]. In this way, all the ubiquitous services can be wrapped as Web service components. The JINI service from Sun Microsystems is also used to create dynamically networked components, software agents, and services that scale from the device to the enterprise. Its unique qualities include (i) code mobility, (ii) leasing, and (iii) integration. However, in contrast to Web services, there is no standard and open architecture-based module for service federations in the JINI service, just as Web services cannot support user-oriented GUIs and interactions. To minimize disadvantages and to maximize advantages of both services, Web services are used for back-end communications, whereas JINI services for front-end communications. Context-aware devices and user-friendly displays are described as JINI objects, which can be easily registered, discovered, and implemented in mobile and context-aware environments.

A Topic Map-based context management provides a foundation for interoperable context-aware service environments where computing entities can easily exchange and interpret contexts based on the explicit context representation. Topic Map is a new ISO standard for describing knowledge structures and associations between them

with information resources[16]. Topic Map consists of three basic concepts: Topics, Associations, and Occurrences. Topics are the most fundamental concept in Topic Map, which defines a subject. A topic may be linked to one or more information resources that are deemed to be relevant to the topic in some way. Such resources are called occurrences of the topic. Associations play a role in describing relationships between topics. Topic Map has a standard query language called Topic Map Query Language (TMQL)[16]. Further, Topic map has the flexibility for explicitly representing contexts, easy merging of multiple contexts based on the Public Subject Indicator (PSI), multiple viewing of contexts based on the scope representation, and standard-based querying and reasoning of contexts.

### 3 Context-Aware and Augmented Reality-Based Ubiquitous Services

This section explains how ubiquitous contexts are managed, queried, and reasoned to provide more relevant and human-oriented services. It also discusses how to utilize augmented reality for executing context-aware collaborations, which gives more natural and intuitive interactions and interfaces.

#### 3.1 U-Context Management

By representing contexts for ubiquitous services and collaborations as easily interpreted semantic ontologies, the context-aware service framework enables intelligent applications to retrieve contexts using declarative queries and supports the inference of higher-level contexts from the basic contexts [14]. The proposed context-related infrastructure consists of several context-aware collaborating components as shown in Fig. 1 and 2. In particular, the context acquisition and maintenance module discovers and gathers contexts from mobile devices such as PDA and cellular phone, RFIDs, and Bluetooth-enabled sensors. Then, it asserts the gathered contexts into the context knowledge base. The context knowledge base also stores context ontologies given by users and related services. It links the context ontology and contexts in a single semantic model and provides interfaces for the context query engine and context reasoning engine to manipulate correlated contexts. The context query engine provides an abstract interface for applications to extract desired contexts from the knowledge base. In particular, contexts are maintained in three different levels of details adaptable to a dynamically changing environment: 1) proxy generation for each context, 2) cluster generation based on the hierarchical representation of contexts among persons, devices, and U-service proxies, and 3) Topic Map-based integrated context map generation from the registered proxies and clusters.

In order for each context to be easily registered, queried, and discovered over U-CAFÉ, it is wrapped as a JINI proxy with a context wrapper. Thus, it is possible for the service requestor to find the registered proxy with the help of the JINI lookup service. The Topic Map-based context wrapper attached to each proxy plays another important role in matching semantics and searching for context relations in the U-CAFÉ service network. There are several context wrappers according to the type of contexts such as persons, devices, locations, and U-services.

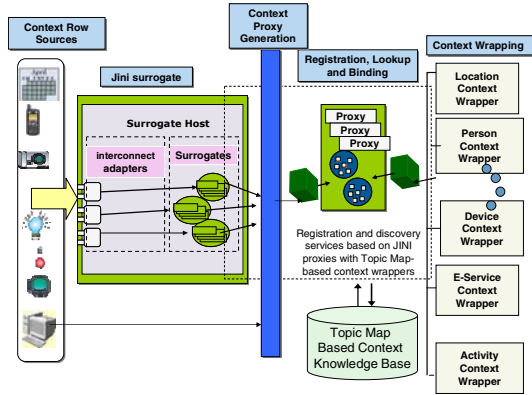


Fig. 2. Context acquisition and maintenance

Moreover, clusters are constructed and maintained based on the hierarchical representation of contexts among persons, devices, and E-services as shown in Fig. 3(a). The hierarchy consists of four layers. The bottom layer includes a range of mobile and fixed devices; neither hardware architecture nor operating system must be homogeneous [1]. The second layer contains device proxies, which every device has. The third layer is the user-proxy layer. Every user in the U-CAFÉ service network has a personal user proxy. This layer can store applications and a user’s state. The fourth layer is the U-service layer, where the architecture provides shared engineering applications, utilities, and servers.

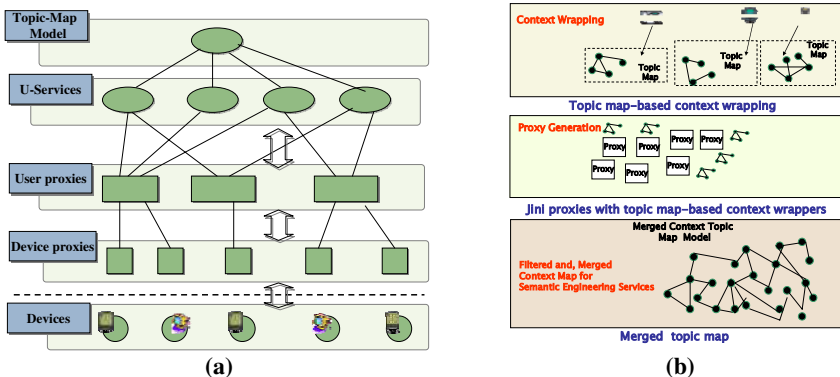


Fig. 3. Hierarchical context representation in U-CAFÉ: (a) hierarchical representation of context relations among users, devices, and E-services and (b) Topic Map-based context merging

Finally, registered proxies and clusters are merged to a Topic Map-based integrated context map as shown in Fig. 3(b). One of the main reasons in representing three different levels of details about contexts is to provide multiple views of the contexts. In addition, it makes it possible to enhance querying and reasoning contexts. Thus,

this representation can make ubiquitous services be more adoptable to a dynamic changing environment.

However, some sensors or devices such as PDAs and cellular phones cannot support the JINI services due to hardware and software limitations. To overcome these kinds of limitations, we implemented JINI surrogate services for those devices as shown in Fig. 2. A surrogate is a facilitator that enables for a device that cannot run over the JINI service network to communicate with registered proxies over the JINI service network. Thus, it is possible to consistently maintain not only JINI interoperable contexts but also JINI non-interoperable contexts using surrogates regardless of device limitations.

### 3.2 Augmented Reality-Based Collaborative Interactions

AR is a natural complement to mobile computing, since a mobile system can assist the user directly in several situations. In ubiquitous environments, mobile devices can be utilized for distributed collaboration, where as AR devices can be effectively used for co-located collaboration [2,12]. The AR-based collaboration broker consists of 4 major modules as shown in Fig. 4: 1) U-context interface module, 2) U-service binding module, 3) tracking module, and 4) rendering module.

Internally, the tracking module and rendering module support AR applications. The tracking module is based on a marker-based tracking technique, also supporting multi-marker tracking capabilities. In this research, ARToolkit is utilized [7]. The rendering module embeds the 3D virtual reality of service and context information onto the physical reality image synchronized by the tracking module. Externally, the U-Context interface module and U-Service binding module are used to communicate with the U-Context broker and U-Service broker for context and service information retrieval and synchronization. The U-Service binding module receives virtual models from the U-Service broker, then, applies various interactions, and finally feedbacks the interactions to the U-Service broker, which can modify the original model or generate new models. Similarly, the U-Context interface module gets context information from the U-Context broker and then embeds the contexts to AR, which can move 3D virtual models or transform them. Further, it also feedbacks new contexts generated from AR interactions to the U-Context broker, which bi-augments each other.

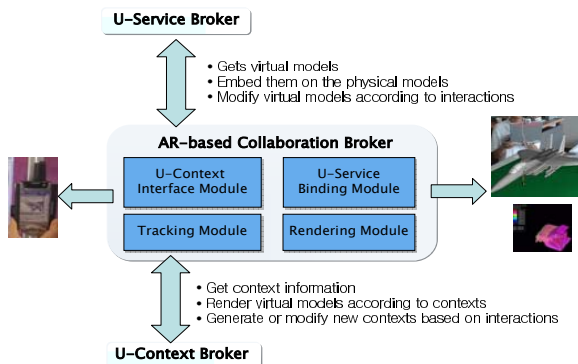


Fig. 4. Modules for AR-based collaboration

## 4 System Implementation

This section explains how the proposed framework can be integrated and applied to various ubiquitous and context-aware applications. To illustrate the benefits of U-CAFÉ, we present the following two scenarios: 1) ubiquitous collaboration for 3D product design and 2) AR-supported ubiquitous home. They show how context awareness is used to relate users with devices and to manage their relations as proxies, and show how context awareness is utilized for various collaborations.

We assume that a user, Mr. CNU, requested an engineering service via mobile device at CisLab. During the analysis of the result, the U-CAFÉ service network advises him to investigate the analysis result with a large display or AR device instead of his PDA due to hardware and software limitations. While analyzing the result using the AR-based interface, unfortunately, he finds a serious problem and, thus, he wishes to collaborate with Mr. B for resolving the problem. However, Mr. B can only access his PDA with which he cannot visualize a large 3D model. Thus, they cannot collaborate with each other based on the previous engineering collaboration environment.

U-CAFÉ manages all the contexts related to this scenario such as device and persons using Topic Map as shown in Fig. 5 when Mr. CNU enters CisLab. It describes topics related to Mr. CNU and Cis-PDA, and their associations.

```

<topic id="Mr.CNU">
  <instanceOf>
    <topicRef xlink:href="#Person"/>
  </instanceOf>
  <baseName>
    <baseNameString>
      Mr. A
    </baseNameString>
  </baseName>
</topic>

<topic id="Cis-PDA">
  <instanceOf>
    <topicRef xlink:href="#Mobile-Device"/>
  </instanceOf>
  <baseName>
    <baseNameString>CISPDA</baseNameString>
  </baseName>
  <occurrence>
    <instanceOf>
      <topicRef xlink:href="#BluetoothMAC"/>
    </instanceOf>
    <resourceData>168.131.132.123</resourceData>
  </occurrence>
  </topic>

<topic id="carrying ">
  <baseName>
    <baseNameString>carrying</baseNameString>
  </baseName>
</topic>

<association id="Mr.CNU-carrying-Cis-PDA">
  <instanceOf>
    <topicRef xlink:href="#carrying"/>
  </instanceOf>
  <member>
    <roleSpec>
      <topicRef xlink:href="#Person"/>
    </roleSpec>
  </member>
  <member>
    <roleSpec>
      <topicRef xlink:href="#Mr.CNU"/>
    </roleSpec>
  </member>
</association>

  <member>
    <roleSpec>
      <topicRef xlink:href="#Device"/>
    </roleSpec>
  </member>
  <member>
    <roleSpec>
      <topicRef xlink:href="#Cis-PDA"/>
    </roleSpec>
  </member>
</association>

<l-Association derived when Mr. CNU enters CisLab-->

<association id="Mr.CNU-locatedIn-CisLab">
  <instanceOf>
    <topicRef xlink:href="#locatedIn"/>
  </instanceOf>
  <member>
    <roleSpec>
      <topicRef xlink:href="#Person"/>
    </roleSpec>
  </member>
  <member>
    <roleSpec>
      <topicRef xlink:href="#Mr.CNU"/>
    </roleSpec>
  </member>
  <member>
    <roleSpec>
      <topicRef xlink:href="#Location"/>
    </roleSpec>
  </member>
  <member>
    <roleSpec>
      <topicRef xlink:href="#CisLab"/>
    </roleSpec>
  </member>
</association>

<association id="CISPDA-locatedIn-CisLab">
  <instanceOf>
    <topicRef xlink:href="#locatedIn"/>
  </instanceOf>
  <member>
    <roleSpec>
      <topicRef xlink:href="#Device"/>
    </roleSpec>
  </member>
  <member>
    <roleSpec>
      <topicRef xlink:href="#Cis-PDA"/>
    </roleSpec>
  </member>
  <member>
    <roleSpec>
      <topicRef xlink:href="#Location"/>
    </roleSpec>
  </member>
  <member>
    <roleSpec>
      <topicRef xlink:href="#CisLab"/>
    </roleSpec>
  </member>
</association>

```

Fig. 5. Some of derived Topic Map-based contexts relating to Mr. CNU when he enters CisLab



Based on these contexts, the following querying and inferencing results based on TMQL show how contexts can be effectively used for providing the right service in the given situation. The following query implies that “find Mr. CNU’s requested ubiquitous services and their status that can be served by the working computers located in CisLab”.

- Query

```
select $R-Service, $Status from
  e-requestedBy($R-Service:Requested-E-Service, CNU:Person),
  e-statusOf($R-Service:Requested-U-Service, $Status:U-Service-Status),
  e-instanceOf($R-Service:Requested-U-Service, $P-Service:Provided-U-Service),
  servedBy($P-Service:Provided-U-Service, $Computer:Device),
  locatedIn($Computer:Device, CisLab:Location),
  conditionOf($Computer:Device, GoodCondition:Condition) ?
```

- Query result

\$R-Service	\$Status
CNU’s R-Service1	U-Service-Finished
CNU’s R-Service2	U-Service-Processing
CNU’s R-Service3	U-Service-Finished

In addition to the context query, it is necessary to infer new contexts from the existing contexts. The following rule can be defined and from which we can derive new contexts. The following rule implies that when Mr. CNU’s requested services are finished, it is necessary to suggest the most desirable device or computer for analyzing the service result.

- Rule

```
e-recommendedWith($R-Service, $Display) :-
  e-requestedBy($R-Service:Requested-U-Service, CNU:Person),
  e-statusOf($R-Service:Requested-U-Service, U-Service-Finished:U-Service-Status),
  e-instanceOf($R-Service:Requested-U-Service, $P-Service:Provided-U-Service),
  desiredBy($P-Service:Provided-U-Service, $Display:Device-Condition).
```

Fig. 6 shows how Mr. CNU and B can collaborate with each other on a mobile device regardless of inhomogeneous hardware and software platforms and system limitations and how the AR-based technique can be applied to construct an intelligent home. Fig. 6(a)-(c) how a context aware-based application level of details is applied to suggest a possible solution. Note that this kind of collaboration is impossible in the existing concept of engineering collaboration. Fig. 6(d)-(f) show how an AR technique and context awareness are applied to an intelligent home. Usually, it costs too much to build a well-equipped intelligent home with various sensors and devices. Thus, many researchers just assume that the intelligent space has been virtually constructed although it has not been, and then they just develop softwares and test them for assuring their ideas and approaches. Theoretically, it works quite well, but it is doubtful that it would work well in the physical environment.

To verify the effectiveness of utilizing AR on ubiquitous systems, we constructed a miniaturized intelligent home. We attached various kinds of sensors such as lights, motors, infrared sensors, and temperature sensors. But, we realized that it is quite difficult to embed moving or dynamic objects to this environment, which limits constructing realistic context-aware experiments. Using the AR technique, however, we can generate various dynamic virtual models that would be embedded to the physical model. For example, Fig. 6(d) and (e) show a miniaturized intelligent room before and after the AR technique has been applied. Fig. 6(f) shows that the light is turned on when a person enters the kitchen. In other words, when a virtual person comes into the kitchen, the system can automatically turn on the light by calculating the location of the virtual person and requesting the E-Context Broker for generating location

contexts, which orders the intelligent home manager to turn on the light. Note that such a ubiquitous environment can be much more realistic and scalable if the AR technique can be fully utilized.



**Fig. 6.** AR applications for collaborative interactions and intelligent home

## 5 Conclusion

U-CAFÉ has been proposed for supporting various ubiquitous applications such as engineering services and intelligent home. The proposed approach adopted a semantic web-based context management for supporting various ubiquitous services. Further, augmented reality has been used to realize more human-centric interfaces in which three-dimensional computer graphics are superimposed over real objects. We realized the AR-based context aware techniques can be very effective interfaces for collaborations: 1) seamless interaction between real and virtual environments, 2) the ability to enhance the reality, 3) the presence of spatial cues for various kinds of collaboration such as product development and intelligent home, and 4) the ability to transition smoothly between reality and virtuality.

Several areas of research related to U-CAFÉ still remain. There is a need to develop a formal representation and service acquisition strategies. Further, hybrid interfaces that integrate AR technology with other collaborative techniques need to be further explored.

## Acknowledgement

This work was supported by grant No. (R01-2003-000-10171-0) from the Basic Research Program of the Korea Science & Engineering Foundation.

## References

1. Anhalt, J., Smailagic, A., Siewiorek, D.P., Gemperle, F., Salber, D., Weber, S., Beck, J., Jennings, J.: Toward context-aware computing: experiences and lessons. *IEEE Intelligent Systems* 16(2001) 38-46
2. Billinghurst, M. Kato, H.: Collaborative augmented reality. *Communications of the ACM* 45(2002) 64-70
3. BPEL4WS, <http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/> (2003)
4. Brumitt, B., Meyers, B., Krumm, J., Kern, A., Shafer, S.: EasyLiving: technologies for intelligent environments. *Proc. 2<sup>nd</sup> Int'l Symp. Handled and Ubiquitous Computing(HUC2000)*, LNCS 1927, Springer-Verlag (2000) 12-29
5. Chen, H., Finin, T., Joshi, A., Kagal, L., Perich, F., Chakraborty, D.: Intelligent agents meet the semantic web in smart spaces. *IEEE Internet Computing* 8(2004) 69-79
6. JINI, <http://www.jini.org/> (2004)
7. Kato, H., Billinghurst, M., Poupyrev, K., Imamoto, K., Tachibana, K.: Virtual object manipulation on a table-top AR environment. *Proc. International Symposium on Augmented Reality* (2000) 111-119
8. Kindberg, T., et al.: People, places, things: web presence for the real world, *Mobile Networks and Applications*. Kluwer Academic Publishers (2002) 365-376
9. Lee, J.Y.: Shape representation and interoperability for virtual prototyping in a distributed design environment. *International Journal of Advanced Manufacturing Technology*, 17(2001) 425-434
10. Lee, J.Y., Lee, S., Kim, H., Kim, H.: A service-oriented approach to engineering web services. *Proc. 11th ISPE International Conf. on Concurrent Engineering: Research and Applications Beijing* (2004) 81-88
11. OWL, <http://www.w3.org/TR/owl-guide/> (2004)
12. Schmalstieg, D., Fuhrmann, A., Hesina, G., Szalavari, Z., Encarnacao, L.M., Gervautz, M., Purgathofer, W.: The studierstube augmented reality project. *Presence: Teleoperators and Virtual Environments* 11 (2002) 33-54
13. Wagner, D., Pintaric, T., Ledermann, F., Schmalstieg, D.: Towards massively multi-user augmented reality on handheld devices. *Pervasive 2005-LNCS 3468* (2005) 208-219
14. Wang, X., Dong, J.S., Chin, C.Y., Semantic space: an infrastructure for smart spaces. *IEEE Pervasive Computing* 3(2004) 32-39
15. Web Services, <http://www.w3.org/2002/ws/> (2002)
16. XML Topic Maps 1.0, <http://www.topicmaps.org/xtm/1.0/> (2001)

# A Smart Method of Cooperative Learning Including Distant Lectures and Its Experimental Evaluations

Dilmurat Tilwaldi<sup>1</sup>, Toshiya Takahashi<sup>1</sup>, Yuichiro Mishima<sup>1</sup>,  
Jun Sawamoto<sup>2</sup>, and Hisao Koizumi<sup>1</sup>

<sup>1</sup> Graduate School of Science and Engineering, Tokyo Denki University

<sup>2</sup> Mitsubishi Electric Corporation

**Abstract.** Cooperative learning links students together to help them attain their learning objective under the guidance of teachers. With the spread of the Internet, cooperative learning has attracted educational technology interest, and research is proceeding into the use of bulletin boards, teleconferencing and chat rooms, etc., to support cooperative learning systems. However, cooperative learning systems tend to be used independently of distance learning, and very little work is being done on cooperative learning methods that include distant lectures. We believed that if remote lectures could be included in cooperative learning by students, the effectiveness of this cooperative learning could be improved. The article proposes a methodology for incorporating distant lectures in cooperative learning and provides an experimental evaluation. To evaluate this method we created a cooperative learning prototype and performed evaluations within our department.

## 1 Introduction

Distance learning may be divided into one of two categories: studies in which students access teaching materials asynchronously via the Internet, and those in which students interact directly with teachers at distant locations synchronously via teleconferencing functions. Cooperative learning links students together to help them attain their learning objective under the guidance of teachers. With the spread of the Internet, cooperative learning has attracted educational technology interest, and research is proceeding into the use of bulletin boards, teleconferencing and chat rooms, etc., to support cooperative learning systems.[1][2][3] This research frequently concentrates on the use of the worldwide web to grasp and control the progress of conversations and discussions.[4][5] However, cooperative learning systems tend to be used independently of distance learning, and very little work is being done on cooperative learning methods that include distant lectures.

The authors have proposed a distant-lecture system in which teaching materials are sent in advance to computers on the student side and teachers send editing commands to display the teaching materials to students.[6][7][8] We believed that if remote lectures could be included in cooperative learning by students, the effectiveness of this cooperative learning could be improved. The article proposes a methodology for incorporating distant lectures in cooperative learning and provides an experimental evaluation.

To evaluate this method we created a cooperative learning prototype and performed evaluations within our department. The experiment was performed twice: the first time was to confirm and evaluate the basic functions of the system. Various problems centering around the system’s chat functions were identified, and the result of the evaluation was unsatisfactory. The second experimental evaluation was performed after improvements to the system’s chat functions, and repeated the evaluations addressing each of the problems.

## 2 A Cooperative Learning Method Including Distant Lectures

### 2.1 Basic Approach

The general flow of cooperative learning that includes distant lectures is shown in Fig. 1. At the initial asynchronous learning stage, the teacher sends the teaching materials with contents prepared for the course to the students in advance. The contents of the teaching materials are used when the distant lecture is given, but it is also possible for the students to use them in their preparation.

At the lecture stage, the teacher sends commands to display the teaching materials, and explains the theme of the studies. Those receiving the lecture see displays corresponding to these commands. Once the explanation of a particular topic has been

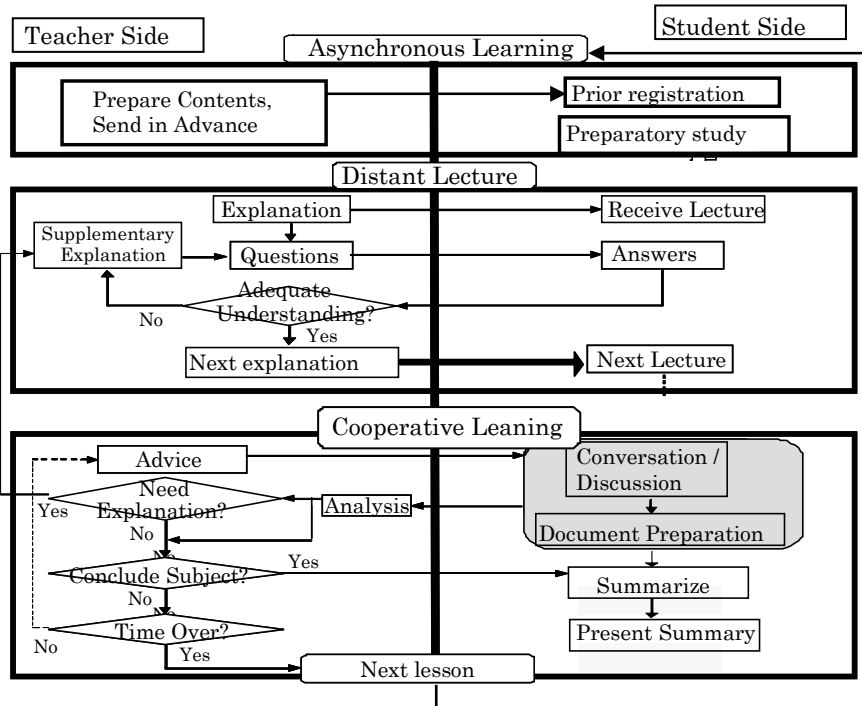


Fig. 1. Flowchart for the Learning Process

completed, the teacher sends a question command and checks the degree of understanding. If teachers feel that understanding is inadequate, they will give additional explanation(s) and check again. When adequate understanding has been achieved, the lecture can proceed. The cooperative learning stage begins once the lecture has been completed.

At the cooperative learning stage, students are divided into groups in which learning proceeds by way of conversations and discussions on the theme of the study and the joint creation of a document. The teacher watches the log of this process for each group, monitors the progress of document creation by group members and gives advice. If further instruction proves necessary, they revert to the distant lecture stage and appropriate additional explanations are given. As soon as the teacher feels that adequate understanding has been achieved by group members, they resume cooperative learning and further instructions are given.

Discussions on the study theme and document creation continue until the group is led to the right conclusions. Once the group has completed the study, the teacher confirms its completion, and the group uses the document(s) to prepare and submit a report. The study as a whole ends with the publication of these reports. The teacher refers to the results of the study, including chat logs and the data on the information shared, in preparing feedback for the next lecture.

### 2.2 Configuration of Cooperative Learning System

The system configuration is shown in Fig. 2. The cooperative learning system consists of the software for editing teaching materials in real time (abbrev. to R/SW below), the chat system, and the information sharing system.

R/SW has functions allowing commands to be sent during the lecture that cause the content of the teaching materials to be displayed to students. In addition to these content display commands, there are also commands allow the teacher to check the degree of student understanding. There are also functions to analyze the results of queries.

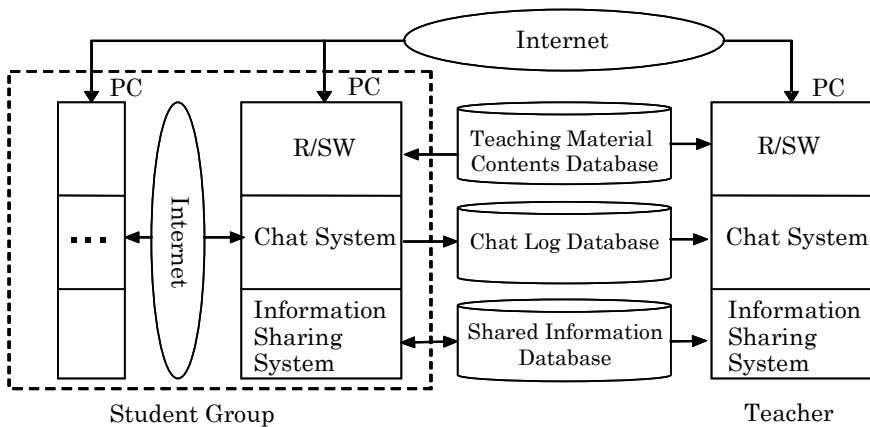


Fig. 2. System Block Diagram

The chat system uses NetMeeting chat, with tags that provide graphic identification of the kind of content that is being sent. This eliminates the misunderstandings that can easily arise in text-only chat sessions, and assists in judging what kind of discussions were being carried out by the group members as they followed the chat log. These tags also facilitate the investigation and analysis of the learning process from the chat log.

Because we considered that discussions would proceed by a repeated process of reaching agreements after exchanges of opinion between the participants, we set up tags to express “agreement,” and “opposition,” with “question” and “explanation” for questions and answers. Since conversations would not be confined to discussions, we also prepared an “other” tag.

We used a Wiki[9] as the system for sharing information. A Wiki is a web contents management system that facilitates access via a web browser, and allows freedom in issuing and editing pages.

### 2.3 System Support

The block diagram of the methods by which support was provided is show in Fig. 3. In order for students to follow the learning theme and achieve the objectives of the study, a document giving the results of their studies is prepared using discussions that proceed via chat and an information-sharing server. In this process, logs are accumulated of the chat conversations and the changes made to the document on the information-sharing server. Real time support during the lecture time is provided by analysis of the data on the server, R/SW is used to grasp the students’ answers to questions, and provides the additional factor of distant lectures that are responsive to the students. It is very important, for the proper support of this kind of learning, that the process of distant cooperative learning should be analyzed, and this is done using the log of conversations and document creation.

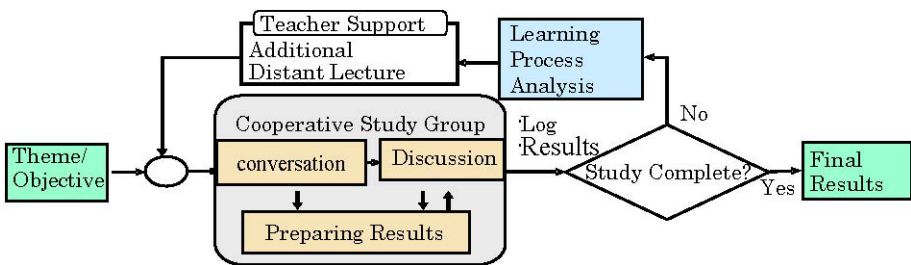


Fig. 3. System Support Methods

### 2.4 Grasping the Students’ Responses

In order for the degree of understanding to be properly assessed in synchronous learning, R/SW commands are used to question understanding and R/SW analytical functions are used on the resulting answers. Here, the degree of student understanding is

assessed on the basis of multiple-choice Q&A forms and single-question answer forms for 20- to 40-word answers. Selection is performed in response to the teacher's questions by means of branching "radio" buttons.

The responses accumulated on the teaching side are checked to grasp the degree of understanding. The answers are totaled, and the average numbers of those selecting a particular branch option are ascertained. This provides a quantitative assessment of the overall level of understanding. The single-question answers are scanned for the keywords that they should contain to assess their correctness. For example, if asked what an interpreter language is, the answer is assumed to be correct if it contains the words "interpret" or "execute."

### **3 Experiments and Evaluation**

#### **3.1 The First Test, Its Evaluation**

##### **3.1.1 The Experimental Environment Within the Department**

The first experimental test was performed with ten fourth-year students in the department. They were divided into two teams, A and B, for the studies. Because the experiment envisages students at distant locations, the students were required not to sit next to one another in the lecture room, and not to communicate face-to-face. All students possessed notebook computers, and were connected via a wireless LAN. The students' computers had pre-installed R/SW, NetMeeting (used for chat) and the Wiki used as the shared information server.

##### **3.1.2 Experimental Schedule and Study Theme**

The experiment was performed over the period May 25 through June 8, 2004, during which there were four sessions, each of 90 minutes. In a fifth and final session, held on June 11, each team announced its results. The theme of the cooperative learning for this experiment was "An Investigation of the Suitability of Distance Learning Using the Internet for the Regions Surrounding the Takla Makan Desert and, if Suitable, of Problems Arising," and after the teacher lectured on the theme of the study, the group commented cooperative learning as discussions continued.

##### **3.1.3 Results and Evaluation**

For the first experiment, we performed analyses of the tags, the chat conversation flow, the Wiki information flow, the distant lecture itself, the questions and answers, the students' announced results, and the results of a questionnaire and opinion survey. As a result, we were able to confirm the overall flow of the cooperative learning process including distant lectures, but the following problems were identified. (1) Too much use was made of the "Other" tag (35.7%), making analysis of the discussion difficult; (2) the widely differing time lags between discussions and document preparation tended to disrupt discussion; (3) it was difficult to know the situation of other students, etc. We accordingly decided that satisfactory results could not be expected if the experiment were to be continued in its current form, and made improvements before carrying out the second experimental test and evaluation.



## 3.2 System Improvements

### 3.2.1 Improved Chat System

For this research, we developed a chat system with tags that would give simultaneous information on the state of discussions and document preparation. We have called this “semantic chat.” A typical semantic chat screen is shown in Fig. 4.

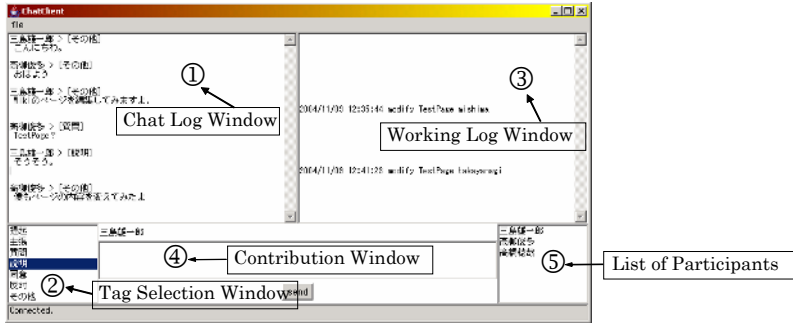


Fig. 4. Typical Semantic Chat Screen

The chat log with tags is shown in window (1), the tag selection list in window (2), the log of document creation in window (3), the area for entering contributions in window (4), and the list of participants in window (5). Meaningful tags are selected from the list in (2), and they are also allocated to function keys for selection and input without having to remove hands from the keyboard.

The chat log is on the left, but in the window on the right, Wiki changes are logged along the same time line. This means that as cooperative distance learning proceeds in parallel with document preparation, the progress of both discussions and preparation can be grasped simultaneously. We believe that this kind of environment, by providing a log of progress in documentation, will enable the teacher to distinguish between those students who are and are not participating in the study, whether or not they are actually participating in the active give-and-take of discussion.

### 3.2.2 Short Supplementary Lectures During Cooperative Learning

In the first experiment, we tried the method of having the students attempt to resolve among themselves any questions arising from the study theme during the cooperative learning process. However, analysis of the content of the discussions and the jointly prepared reports revealed that to some extent supplementary lectures from the teacher were necessary. In the second experiment, we therefore arranged to ascertain the student response and to provide appropriate supplementary instruction.

## 3.3 The Second Test, Its Evaluation and Consideration

### 3.3.1 Test Schedule and Content

In the second experimental test, nine of the department’s fourth-year students participated on two occasions, Feb. 7 and 10, 2005. Nine of the participants in the previous test

were divided into two teams, Team A with four members and Team B with five. The study theme this time was different; “A Review of Policies to Encourage More Japanese Tourists to Visit the Autonomous Uighur Region of Xinjiang Province on China’s Silk Road.” There were two group discussions. One concentrated on what could be done to bring more Japanese tourists to China’s Silk Road and the Autonomous Uighur Region, and the other concentrated on proposing appropriate measures to solve the problems involved. Participants were also pre-tested by E-mail before the experiment.

On February 7, the teacher gave the lecture on the study theme, then checked the degree of understanding, gave a supplementary lecture, and then rechecked understanding.

In the February 10 experiment, cooperative learning took place, with checks of student interest, additional lectures, checks of comprehensive, etc., and finally each team brought its document to a conclusion and submitted a report. The experiment was followed by a test and a questionnaire.

**3.3.2 Experimental Results and Evaluation**

On February 7, the students’ reactions were ascertained after the distant lecture. Check points were displayed on the R/SW communications screen, and student understanding during the distant lecture was assessed by asking questions with the answers to be selected from four alternatives. This check was repeated four times. The results achieved by the students were assessed and additional lectures were given where this appeared to be necessary, and the same four questions were asked to check understanding. A comparison between the students’ understanding before and after the additional lecture is shown in Table 1, from which it is clear that student understanding improved.

On February 10, during the cooperative learning, the teacher analyzed the semantic chat log and the document, and realized where students were experiencing problems and where more detailed explanations were necessary. First, the degree of student interest in this possibility was checked. Table 2 shows the results of two checks of student interest.

**Table 1.** Checks of Understanding Before and After an Additional Lecture

Item	Before				After			
	1	2	3	4	1	2	3	4
No. of checks performed	1	2	3	4	1	2	3	4
No. answering correctly	7	7	8	7	12	11	11	12
No. answering incorrectly	5	4	2	5	0	0	0	0
No. of those not answering	0	1	2	0	0	1	1	0
Percentage of right answers	60.42%				95.83%			

**Table 2.** Degree of Student Interest

Check Necessary (Yes)or Unnecessary (No)	Team A (No.)			Team B (No.)		
	Yes	No	Either	Yes	No	Either
1 <sup>st</sup>	4	0	1	4	0	0
2 <sup>nd</sup>	4	1	0	3	0	1

### 3.3.3 Considerations

#### (1) Log Analysis and Semantic Chat

We compared the frequency with which the “others” tag was selected in the first cooperative learning experiment with that observed when the semantic chat system was used. This comparison is shown in Table 3. The table shows that when the semantic chat system is used, the “others” tag is used some 11.8 percentage points less than when NetMeeting was used. This suggests that the method of associating tags in the semantic chat system has become considerably more convenient, and that the students had become more familiar with their usage.

**Table 3.** Frequency of Tag Usage

Type of Chat System	“Others” (%)	Not “Others” (%)
NetMeeting Chat (1 <sup>st</sup> experiment)	35.7	64.3
Semantic Chat (2 <sup>nd</sup> experiment)	23.9	76.1

A questionnaire survey was carried out after the study. Divided into 15 categories, each with five levels of response, it also allowed for open-ended expressions of opinion. In response to the questions “Was the semantic chat system effective?” “Did the semantic chat window showing the updating of Wiki information prove useful?” and “Was the supplementary lecture helpful?” many students mentioned as positive factors that the semantic chat system was good in that it had simplified the input of tags, and that the Wiki update window facilitated progress by eliminating wasteful confirmations. However, there were also who indicated that although the display of information on the work of other students during the discussions was effective, there was also a need to display the status of the discussions with other students while the work was proceeding.

#### (2) Analysis of Supplementary Lectures

From Table 1 it is clear that the supplementary lecture improved student understanding, and from Table 2 that supplementary lectures are also necessary during cooperative learning. The results of tests of understanding performed after the supplementary lectures show that almost all students had understood their content.

#### (3) Pre-Test and Post-Test Analysis

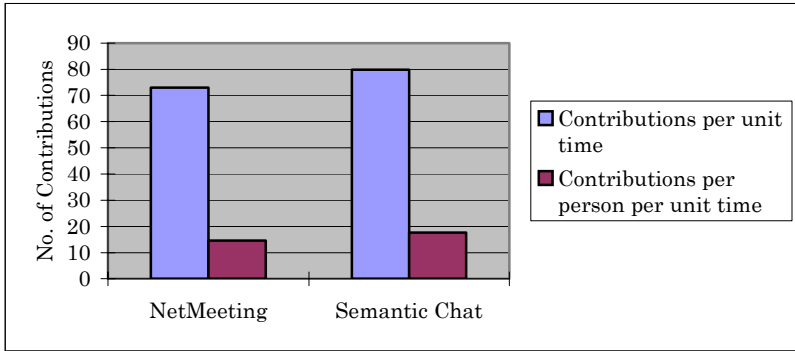
The same test of the study theme was performed both before and after the cooperative learning. Fig. 5 shows the results of the group tests. The average scores show considerable differences before and after the cooperative learning. In terms of actual test scores, all students achieved higher scores.

**Table 4.** Pre-Test and Post-Test Results

Team	Pre-Test	Post-Test	Difference
A	22.75	69.5	+46.74
B	17.33	76.66	+59.33

**(4) A Comparison of NetMeeting and Semantic Chat**

NetMeeting was used for discussions in the first experiment and semantic chat was used in the second. Semantic chat dispensed with the need to enter tags manually, and we expected that this would affect the number of contributions. The number of contributions per unit time in the first and second experiments is shown in Fig. 5.



**Fig. 5.** A Comparison of the Number of Contributions Made Using NetMeeting and Semantic Chat

The comparison of NetMeeting and semantic chat given in Fig. 5 shows that both the total number of contributions to the discussion made per unit time, and the average number of contributions made by individuals per unit time, are both higher for semantic chat. This appears to be because dispensing with the need to enter tags by hand has encouraged more contributions.

**4 Conclusions**

This article proposes a method of cooperative learning that incorporates distant lectures and provides its experimental evaluation. The first experimental evaluation was performed for cooperative learning with the theme “An Investigation of the Suitability of Distance Learning Using the Internet for the Regions Surrounding the Takla Makan Desert and, if Suitable, of Problems Arising.” As a result of analysis of the study process we encountered several problems. To solve these problems we developed the semantic chat system. In the second experimental evaluation, the study theme was “A Review of Policies to Encourage More Japanese Tourists to Visit the Autonomous Uighur Region of Xinjiang Province on China’s Silk Road.” As a result, a new problem of awareness was identified in this method.

Future issues to be resolved are the establishment of the parameters by which the effectiveness of studies performed via cooperative distance learning can be assessed. We also plan further practical testing and improvement of the developed system, and intend to perform experimental evaluation of the time lags between discussion and document preparation.

## References

- [1] The Advanced Learning Infrastructure Consortium (ALIC) (Ed.) “*On Collaborative Learning*”, the annual White Paper on e-Learning 2003/2004 edition, P295, Ohmsha Ltd., Tokyo, 2003. (in Japanese)
- [2] “The Virtual UNIVERSITY”, Edited by Kevin Robins and Frank Webster, OXFORD UNIVERSITY PRESS, 2002.
- [3] “Virtual Environments for Teaching & Learning”, Edited by L.C.Jain etc, World Scientific, 2002
- [4] Akiko INABA etc, “*An Intelligent Supporting of Discussion for the Distributed Cooperative Learning Environment*” Transactions of the Institute of Electronics, Information and Communication Engineers (IEICE), A Vol.j790A No.2, pp207-30, 1996. (in Japanese)
- [5] Keisuke YAGI etc, “*A Novel Distance Learning System for the TIDE Project*” Transactions of the IEICE, D-II Vol.j84 – DII No.6 p.1139, 2001-6. (in Japanese)
- [6] Yuichi MISHIMA, Tomoo INOUE, etc, “*Activity-aware semantic chat system based on a study of learning process in distance collaborative learning*” Information Processing Society of Japan Special Interest Group (IPSJ SIG) on Groupware and Network Services Workshop 2004, pp81-86, 2004. (in Japanese)
- [7] Dilmurat Tilwaldi etc “*A Real-time Editing Method of Teaching Materials in the Unified Synchronous/Asynchronous Distance Learning*” IPSJ SIG Technical Report DPS-113, PP385-386, 2003-6. (in Japanese)
- [8] Toshiya TAKAHASHI etc, “*A Proposal of An Education Support System with Functions of Real-time Editing Contents based on WebCT*” IPSJ Workshop on Multimedia Communications and Distributed Processing, pp143-148, 2004-12. (in Japanese)
- [9] Hisayoshi ITO, “*An Attempt of Information-sharing within a Laboratory on Wiki Clone*” IEICE Technical Report, Vol.103 No.226, pp13-18, 2003-7. (in Japanese)

# u-KoMIPS: A Medical Image Processing System in a Ubiquitous Environment

Soo Jin Lee and Moon Hae Kim

Konkuk University,  
Seoul 143-170, Korea  
{leesj, mhkim}@konkuk.ac.kr

**Abstract.** The ubiquitous computing paradigm has made the telemedicine field being changed. However, research on the telemedicine field in a ubiquitous environment has not been carried out yet. In this paper, we have designed and implemented a ubiquitous medical image processing system for telemedicine applications based on the time-triggered message-triggered object (TMO) structuring scheme that is a distributed real-time object model. The initial version of our system is named KoMIPS (Konkuk Medical Image Processing System) that is a result of joint work with a Samsung Medical Center team. KoMIPS is a stand-alone system. Currently, KoMIPS is being extended to run on a ubiquitous environment. The extended version, named u-KoMIPS (ubiquitous KoMIPS), is designed with the TMO model and its execution environment is based on TMOSM/Linux that is an execution engine for a TMO structured program on Linux. The u-KoMIPS can acquire a medical image from various medical image scanners (e.g., MRI, CT, gamma camera, etc.) and then convert it into a DICOM image, transfer the DICOM image to one or more clients. Then, a client can analyze, process, and diagnose the DICOM image. We expect that telemedicine applications based on our system would accurately acquire a medical image from various medical image devices and efficiently analyze it.

## 1 Introduction

Today, demands on an effective telemedicine system have been ever increasing. With the advances in information technologies and telecommunication technologies, the construction of an effective telemedicine system becomes possible. And embedded systems equipped with microprocessors are widely applied to various fields with advances in information technologies and telecommunication technologies. In addition, the ubiquitous computing paradigm leads application areas of embedded systems to become much broader. Many fields including the telemedicine field are affected by this phenomenon. Now, research and development for various medical systems is being conducted, for example, u-hospital, telemedicine used IMT 2000 service, and web-based PACS. In Europe, mobile telemedicine applications using a cellular phone or a PDA have already been introduced [1-3]. The vigorous progress in telemedicine is being made due to efficient environments such as improvement IT business, inter-

net communication network, PACS, wireless LAN, mobile phone, etc. However, most of current telemedicine systems depend on special platform/devices and those are not compatible each other [4]. Also, research on a ubiquitous telemedicine is not in a mature stage.

In this paper, we propose a ubiquitous telemedicine system by extending KoMIPS (Konkuk Medical Image Processing System) that has been developed in collaboration with a Samsung Medical Center team. KoMIPS consists of a small gamma camera for a breast cancer and medical image processing system. We have remodeled and implemented KoMIPS to a TMO based medical image processing system for telemedicine applications. The TMO model is a distributed real-time object model and briefly introduced in section 2.

Our extended system, named u-KoMIPS (ubiquitous KoMIPS) can acquire a medical image from various medical image devices (such as MRI, CT, and gamma camera) and then converts it into an equivalent DICOM (Digital Image and Communication in Medicine) image that is an international medical image format standard. The u-KoMIPS provides a set of analysis functions such as ROI (region of interest) setting, contrast adjust, MCA (multi-channel analyzer), and a set of image processing functions such as filtering, edge detection, and binary arithmetic.

The rest of this paper is organized as follows: In section 2, we briefly introduce the TMO model and DICOM format that are the basis of our approach. Section 3 presents KoMIPS overview and the TMO based medical image processing system. In section 4, we introduce an application that has been developed applied TMO approach. Finally, section 5 summarizes the paper with future work.

## 2 Background

### 2.1 TMO Scheme

TMO is a natural, syntactically minor, and semantically powerful extension of the conventional Object(s) [5-7]. Particularly, TMO is a high-level real-time computing object. Member functions (i.e., methods) are executed within specified time. Timing requirements are specified naturally intuitive forms with no esoteric styles imposed. As depicted in Figure 1, the basic TMO structure consists of four parts:

- Object Data Store (ODS): the basic unit of storage which can be exclusively accessed by a certain TMO method execution at any given time or shared among concurrent executions of TMO methods (SpMs or SvMs).
- Environment Access Capability (EAC): the list of entry points to remote object methods, logical communication channels, and I/O device interfaces.
- Spontaneous Methods (SpM): a new type of method, also known as the time-triggered (TT) method. The SpM executions are triggered when the real-time clock reaches specific values determined at design time. A SpM has an AAC (Autonomous Activation Condition), which is a specification of the time windows for execution of the SpM.

- Service Method (SvM): conventional service methods. The SvM executions are triggered by service request messages from clients.

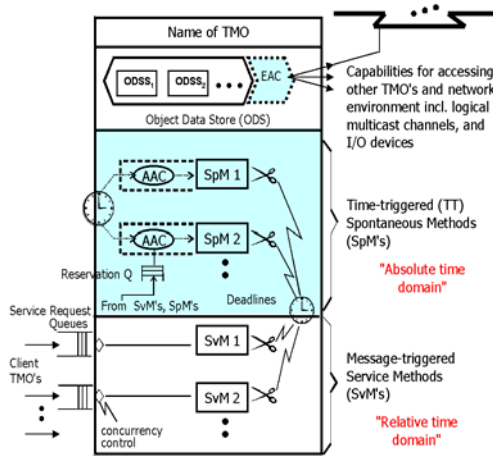


Fig. 1. Structure of TMO (adapted from [5])

## 2.2 Feasibility of TMO for Modeling and Implementing Medical Image Processing System

Fundamental features provided in the TMO programming scheme can enable efficient programming of complex distributed medical image processing system for telemedicine applications. Development of a telemedicine service system requires functionalities of a distributed medical image processing and a medical image processing service. Also, the concept of a service object and provision of high-level API can facilitate the development of a telemedicine system.

First, the TMO scheme provides a sound foundation for programming and executing distributed objects and also the scheme provides various support middleware such as TMOSM/Linux, KelixRT, TMOSM, etc. These features enable a developer to easily design and implement a telemedicine system. Second, the clear separation between SpMs and SvMs, and the BCC rule allow the use of SpM, the time-triggered spontaneous methods, as a means for periodic acquisition of medical signal.

## 2.3 DICOM

The DICOM standard is a specification that describes a means of formatting and exchanging actual image data and associated information such as image calibration and patient's information. The standard applies to the operation of the interface that is used to transfer data in and out of an imaging device such as CT, MRI, X-ray, PET, SPECT, and Ultra-sound [8-10].

Figure 2 shows format of DICOM. DICOM file is a series of "data elements" each of which contains a piece of information.



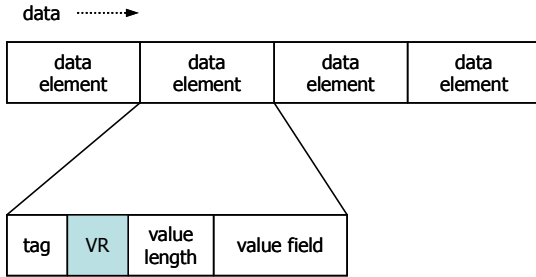


Fig. 2. Format of DICOM

- Data element tag: it contains a piece of information. Each element is described by an “element name” consisting of a pair of 16 bit unsigned integers (“group number”, “data element number”).
- VR (value representation): it is a field that is a 16 bit unsigned character that describes a type of data and a type of data element value.
- Value length: it is a length field that is a 32 bit unsigned even integer that describes the number of bytes from the end of the length field to the beginning of the next data element.

### 3 Design of a Medical Image Processing System Based on the TMO Model

#### 3.1 KoMIPS Overview

KoMIPS is a PC-based cost-effective system that is capable of acquiring, storing, analyzing, and processing medical images. KoMIPS is a result of joint work with a Samsung Medical Center team (department of nuclear medicine) and initial objective of KoMIPS is to facilitate to detect a breast cancer. Figure 3 briefly shows how KoMIPS obtains a medical image. After injection of a radioisotope to a patient, KoMIPS acquires signals using a small gamma camera. Then it constructs, displays, processes, and analyzes an image in real-time. It should be noted that KoMIPS can deal with various types of images although current KoMIPS uses a gamma camera as its scanning device. The gamma camera can be replaced with other scanning devices such as CT (Computer Tomography), MRI (Magnetic Resonance Imaging), SPECT (Single Photon Emission Computerized Tomography), PET (Positron Emission Tomography), etc.

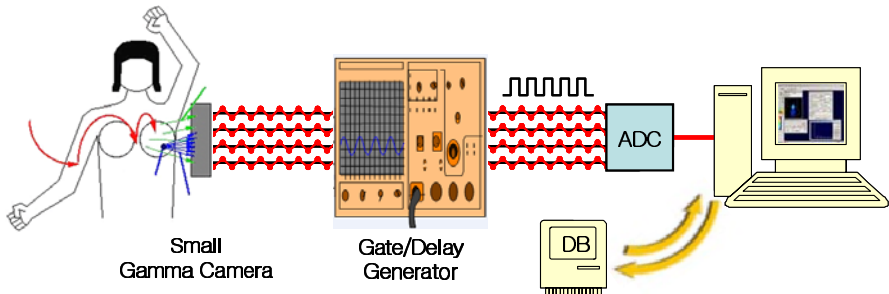


Fig. 3. Overview of KoMIPS

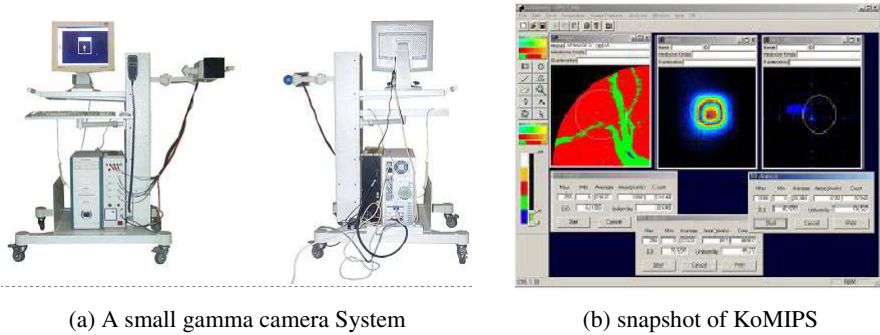


Fig. 4. KoMIPS system

KoMIP can not only load various types of images but also construct images in real-time. Also, KoMIPS converts an acquired raw image into a DICOM image using DICOM encode module and then, stores the image to a database. The Figure 4(a) shows a small gamma camera to acquire a medical image and Figure 4(b) shows a snapshot of image processing in KoMIPS. Several doctors at the Samsung Medical Center have used our KoMIPS prototype extensively and expressed the usefulness of KoMIPS.

### 3.2 System Architecture

With advent of the ubiquitous computing paradigm, research on a telemedicine system is vigorously being progressed. However, research on a ubiquitous telemedicine system is in its beginning stage. Especially, a telemedicine application requiring real-time operations is not being considered yet. In this section, we designed a TMO based medical image processing system, named u-KoMIPS (ubiquitous KoMIPS) running on TMOSM/Linux. TMOSM/Linux is a middleware supporting real-time operations and execution of TMO structured programs.

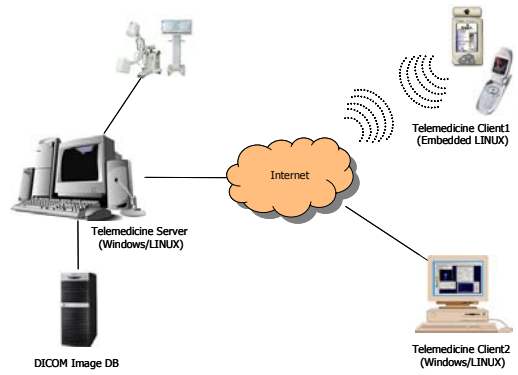


Fig. 5. TMO based medical image processing service

Figure 5 shows a service environment of u-KoMIPS. We installed a (embedded) Linux and a TMOSM/Linux that is a middleware supporting TMO. A client such as desktop, PDA, and mobile phone requests an image or an image processing service of server. A (embedded) Linux, a real-time OS, is enable of rapidly, quickly and seamless information processing.

### 3.3 Modeling of a TMO Based Medical Image Processing System

In this section, we present design of the u-KoMIPS. In the u-KoMIPS, the following objects (TMOs) are used:

- MIAcquisition TMO: an object to acquire a medical image from various medical image scanners such as MRI, CT, SPECT, etc.
- MIProcess TMO: an object to process a medical image.
- MIAalysis TMO: an object to analyze a medical image.
- MIDicom TMO: an object to convert an acquired medical image into a DICOM image.
- MIDisplay TMO: an object to display DICOM/Raw images on mobile information devices such as a PDA, a cellular phone, a HPC, etc.
- MISManager TMO: an object to overall manage requirement of MIService TMO. MIService TMO request MISManager TMO of a DICOM image or image processing service. Then, MISManager TMO requests to a corresponding TMOs.
- MIService TMO: an object to request an image or image processing service to a corresponding server.

Figure 6 presents the relationship among the above mentioned TMOs. SpMs in each TMO perform the roles assigned to the TMO. Interactions among TMOs are

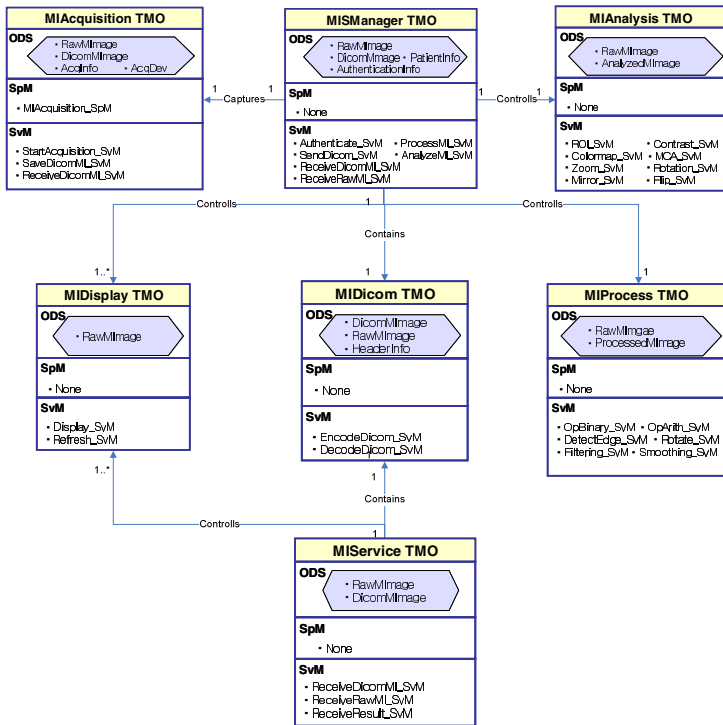


Fig. 6. TMO class diagram for a medical image processing system

occurred through SvMs and EAC. With MIAcquisition TMO, we explain how the roles of a TMO can be performed by SpMs in that TMO and how a TMO can interact with other TMOs.

MIAcquisition TMO is a service object that inputs/outputs medical image data from various medical image scanners. Among the methods in MIAcquisition TMO, MIAcquisition\_SpM acquires raw image data from medical image scanners periodically and saves into a corresponding ODS. In the following, we describe the methods in MIAcquisition TMO in detail.

- MIAcquisition\_SpM: MIAcquisition\_SpM is automatically activated by AAC allocated dynamically in runtime, periodically acquires image signals from medical image scanner and saves them into ODS. At this point, depending on a medical image scanner, various image composition algorithms are selectively applied. Also MIAcquisition\_SpM notifies an acquisition status to MISManager TMO. Specification of temporal conditions in AAC for MIAcquisition\_SpM at design time so as follows:

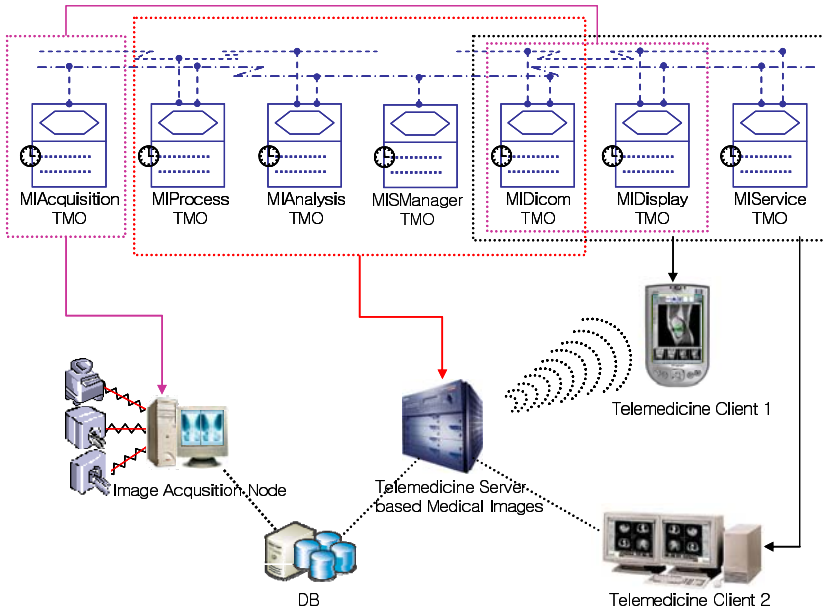
*for*  $T =$  *from*  $TMO\_START + S$   
*to*  $TMO\_START + S + P$   
*every*  $AQ$   
*start-during*  $(T, T + OS\_DELAY\_FOR\_ACQUISITION)$   
*finish-by*  $(T + DEADLINE\_FOR\_ACQUISITION)$

The  $S$  denotes start time that is the time to acquire signal from medical image scanner, and  $P$  denotes stop time that is an end of an acquisition time. The  $AQ$  denotes the time that is request time to compose a medical image, and  $D$  denotes deadline time of signal processing. A SpM with the above AAC starts at the time  $TMO\_START + S$  and is executed until the time  $TMO\_START + S + P$  with the period  $AQ$ . Also, the SpM must start between  $T$  and  $T + OS\_DELAY\_FOR\_ACQUISITION$  and complete its task within  $T + DEADLINE\_FOR\_ACQUISITION$ .  $T$  is a time variable and  $TMO\_START$  refers to the start time of the TMO execution engine.  $OS\_DELAY\_FOR\_ACQUISITION$  is the time spent by OS for activating a SpM.

- StartAcquisition\_SvM: a method that is called by MISManager TMO in case of receiving request of an image acquisition from client.
- SaveDicomMI\_SvM: a method that is called in case of saving a DICOM image to medical image database. SaveDicomMI\_SvM reads a DICOM image data from ODS and saves it DB.
- ReceiveDicomMI\_SvM: a method that is called by MIDicom TMO in case of trying to send a DICOM image. ReceiveDicomMI\_SvM receives a DICOM image data from MIDicom TMO and writes it to ODS.

## 4 A Prototype of u-KoMIPS

In this section, we present a prototype of the u-KoMIPS. In the prototype, a small gamma camera, a medical image device, is used to acquire a medical image and is applied anger logic algorithm to compose image.



**Fig. 7.** Design architecture of u-KoMIPS

As depicted in Figure 7, the image acquisition node that acquires image signals from various medical image scanners contains MIAcquisition TMO, MIDicom TMO, and MIDisplay TMO. The Telemedicine server is composed of MIPProcess TMO, MIAAnalysis TMO, MISManager TMO, and MIDicom TMO. Also, the client is composed of MIDicom TMO, MIDisplay TMO, and MISService TMO. Client requests a medical image or image processing service of telemedicine server with MISService TMO. Then, MISManager TMO received request from client responses it.

Development environment of the u-KoMIPS as follows:

- Operating System : Linux/Embedded Linux
- Middleware : TMOSM/Linux
- ADC: DAQ Board in National Instrument
- Libraries : Nidaq32.lib, Nidex32.lib
- Development Tool : GTK+, QT
- Database : mySQL

Figure 8 shows a GUI of the DICOM Image Viewer, a part of the u-KoMIPS. As depicted in Figure 8, frequently used functions are provided in a form of ICON. Also, at image loading time, the system analyzes the DICOM v3.0 header and displays the header information with image. In addition, the viewer has an image processing tool bar that can be docked [11]. Figure 8(a) shows the server-side GUI that displays DICOM images such as MRI, CT, Gamma, etc. and analyzes images using color map, ROI, and contrast functions. Figure 8(b) shows transfer of an image in a DICOM image database to client. Figure 8(c) shows the client-side GUI. We have implemented server-side and client-side GUIs using QT.

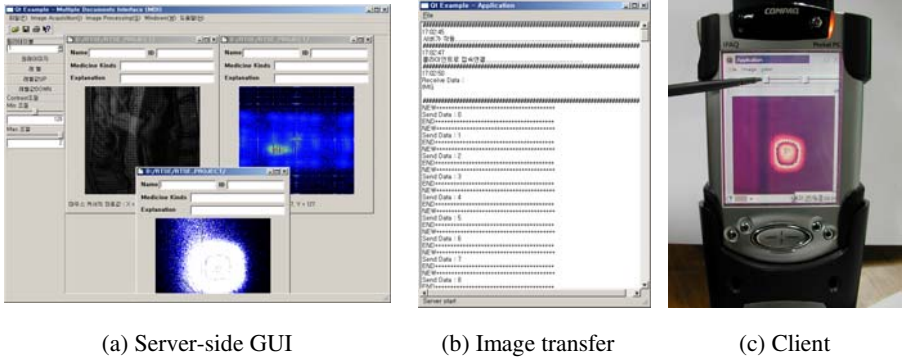


Fig. 8. Prototype of u-KoMIPS

Functions provided with the prototype can be summarized as follows:

- Image acquisition: obtaining raw image data from various sources.
- DICOM encoding/decoding: all images are converted to the corresponding DICOM v3.0 images.
- ROI analysis/copy/paste/save: once ROI is set, then ROI can be copied, pasted, analyzed quantitatively, and saved for future use, e.g., applying saved ROI to same or different images.
- MCA (Multi-Channel Analyzer): for each input channel, the energy (in fact, voltage) of each signal is measured. With signals from all channels, an energy spectrum image can be constructed. This function can improve image analysis capability by removing unnecessary signals and noises prior to the construction of an energy spectrum image.
- Color map control
- Contrast control
- Image processing
  - Flip/mirror
  - Zoom in/out
  - Rotation (0° to 360°)
- Arithmetic/binary operation: image pre-processing such as add, subtract, multiply, divide, AND, OR, and XOR with two images.

## 5 Conclusions

In this paper, we present the design of medical image processing system based on the TMO model for telemedicine applications and implemented a prototype running on PCs and PDAs under the (Embedded) Linux OS. The prototype provides various functions of image processing, analysis, transfer, archive and acquisition. The initial version of our system is named KoMIPS that is a result of joint work with a team in the department of nuclear medicine of the Samsung Medical Center. The current system, named u-KoMIPS, is an extension of KoMIPS of which purpose is to run on a ubiquitous environment.

Our system provides a way to easily develop telemedicine applications not only in a stand-alone system but also in a ubiquitous/distributed environment. Also, our system can enable mobile clients such as PDA to be used in diagnosing a medical image anytime anywhere.

**Acknowledgement.** This research was supported by the MIC (Ministry of Information and Communication), Korea, under the University ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment).

## References

1. B. Woodward, R.S.H. Istepanian and C.I. Richards : Design of a Telemedicine System Using a Mobile Telephone, *IEEE Trans. On Inf. Tech. in Biomed.*, Vol. 5, No. 1, March 2001
2. Negel H. Lovell, F. Magrabi, B.G. Celler, K. Huynh and H. Garsden : Web-based Acquisition, Storage, and Retrieval of Biomedical Signals, *IEEE Engineering in Medicine and Biology*, May/June 2001, pp.38-44
3. J.K. Pollard, S. Rohman and M.E. Fry : A Web-Based Mobile Medical Monitoring System, *International Workshop on Intelligent Data Acquisition and Advanced Computing Systems, Technology and Applications*, 1-4 July 2001, Foros, Ukraine, pp. 32-35
4. Shawn, et al, "Design and implementation of an Internet-based medical image viewing system", *The journal of Systems and Software* 66, (2003) 167-181
5. Kim, K.H.: APIs for Real-Time Distributed Object Programming. *IEEE computer*, (2000) 72-80
6. Kim, K.H., Ishida, M., and Liu, J.: An Efficient Middleware Architecture Supporting Time-Triggered Message-Triggered Objects and an NT-based Implementation. *ISORC*, (1999) 54-63
7. Kim, H.J., Park, S.H., Kim, J.G., and Kim, M.H.: TMO-Linux: A Linux-based Real-time Operating System Supporting Execution of TMOs, *ISORC*, (2002)
8. NEMA Standards Publications PS 3.x, "Digital Imaging and Communications in Medicine (DICOM)", National Electrical manufacturers Association, 1998
9. S.C. Horii, et al., "DICOM: An Introduction to the Standard", [http://www.xray.hmc.psu.edu/dicom/dicom\\_intro/index.html](http://www.xray.hmc.psu.edu/dicom/dicom_intro/index.html), 1997
10. B.A. Levine, et al., "Challenges encountered while implementing a multi-vendor tele-radiology network using DICOM 3.0", *Proc. SPIE* 3035, 1997, pp. 237-246
11. Randy Crane, *A simplified approach to Image Processing*, Prentice Hall, 1997

# The Extended PARLAY X for an Adaptive Context-Aware Personalized Service in a Ubiquitous Computing Environment\*

Sungjune Hong<sup>1</sup>, Sunyoung Han<sup>2,\*\*</sup>, and Kwanho Song<sup>3</sup>

<sup>1</sup> Department of Information and Communication, Yeojoo Institute of Technology,  
454-5 Yeojoo-goon, Kyungki-do 469-800, Korea  
sjhong@mail.yeojoo.ac.kr

<sup>2</sup> Department of Computer Science and Engineering, Konkuk University,  
1, Whayang-Dong, Kwagjin-Gu, Seoul 143-701, Korea  
syhan@kkucc.konkuk.ac.kr

<sup>3</sup> National Internet Development Agency of Korea,  
3F, 1321-11 Secho-2 Dong, Secho-Gu, Seoul, 135-875, Korea  
khsong@nida.or.kr

**Abstract.** This paper describes the extended PARLAY X for the Adaptive Context-aware Personalized Service (ACPS) in a ubiquitous computing environment. It can be expected that the context-awareness, adaptation and personalization for the Quality of Service (QoS) / Quality of Experience (QoE) in a ubiquitous computing environment will be deployed. But the existing PARLAY X is lacking when considering QoS / QoE in network. To address this issue, this paper suggests the extended PARLAY X for ACPS. The objective of this paper is to support the architecture and the Application Programming Interface (API) of the network service for the context-awareness, adaptation and personalization in a ubiquitous computing environment. ACPS provides a user with QoS / QoE in network according to the detected context such as location, speed and user's preference. The architecture of the extended PARLAY X for ACPS is comprised of a Service Creation Environment (SCE), the semantic context broker, and the overlay network. SCE uses Model Driven Architecture (MDA)-based Unified Modeling Language (UML) / Object Constraint Language (OCL) for an expression of context-awareness, adaptation, and personalization. The semantic context broker is a broker role between the SCE and PARLAY gateways. The overlay network is a broker role for QoS / QoE between PARLAY gateway and the IP network.

## 1 Introduction

There is increasing interest in a ubiquitous computing environment with Next Generation Network (NGN). A ubiquitous computing environment with NGN needs

---

\* This work is supported by National Computerization Agency of Korea (NCA) for Research about "Development and Test of Overlay Multicast Transform Device between IPv6 and IPv4" Project in 2005.

\*\* Corresponding author.



the provision of seamless applications in the face of changing value chains and business models, requiring the ongoing replacement and extension of service delivery platform enabled by new information technology and software tools.

New open network service delivery platform standards, such as PARLAY [1] Application Programming Interface (API), are based on the principle of service programming support with network protocol abstraction and the exploitation of state of the art information technology. In addition to PARLAY, the most innovative software development approach, the Model Driven Architecture (MDA) [2], aims to provide total freedom to application development. Consequently, it seems logical to use MDA-based PARLAY such as medini [3] for the rapid and highly automated development of network service on the PARLAY based service delivery platform.

However, the existing MDA-based PARLAY does not consider that the context-awareness [4], adaptation [5], and personalization [6] for Quality of Service (QoS) / Quality of Experience (QoE) [7] in a ubiquitous computing environment. It can be expected QoS / QoE for the customized network service in a ubiquitous computing environment will be deployed. To solve this issue, Web Architecture for Service Platforms (WASP), developed by Telematica Instituut and Ericsson, conducted a research project for context-aware middleware focused on semantic web service [8] technology, using PARLAY to 3G network. WASP focused on semantic web service technology which creates difficulties for many developers who are not adept with the semantic web service in developing a new network service. In addition, WASP lacks in considering adaptation and personalization in a ubiquitous computing environment.

Therefore, this paper suggests the extended PARLAY X for the Adaptive Context-aware Personalized Service (ACPS) for the context-awareness, the adaptation and personalization in a ubiquitous computing environment. All references to 'the extended PARLAY X for ACPS' from this point forward is abbreviated as 'ACPS'.

The objective of this paper is as follows:

- To support the context-awareness, the adaptation, and personalization for the QoS / QoE in a ubiquitous computing environment.

ACPS provides users with the QoS / QoE according to the changing context constraints and the user's preference. The existing PARLAY is the open Application Programming Interface (API) to converse telecommunication, Information Technology (IT), the Internet and new programming paradigm. PARLAY Group, a group of operators, vendors, and IT companies, started in 1998 with the definition of an open network Parlay API. This API is inherently based on an object-oriented technology and the idea is to allow third party application providers to make use of the network (i.e., have value added service interfaces). MDA is an approach to the full lifecycle integration and interoperability of enterprise system comprising of software, hardware, people, and business practices. It provides a systematic framework to understand, design, operate, and evolve all aspects of such enterprise systems, using engineering methods and tools. MDA uses Unified Modeling Language (UML) / Object Constraint Language (OCL) [9]. OCL is a UML extension for expression of semantics.

This paper describes the design and implementation of ACPS in a ubiquitous computing environment, and is paper is organized as follows: Section 2 illustrates the design of ACPS; section 3 describes the implementation of ACPS; Section 4 compares the features and performance of ACPS. Finally, section 5 presents the concluding remarks.

## 2 The Design of ACPS

### 2.1 The Overview of ACPS

A scenario using ACPS is depicted below. We assume that there are the contexts such as location and speed in the surroundings of a wireless device that is detected from sensors called motes [11] or Global Positioning System (GPS). Moreover, we assume that the Wireless LAN region is enough of network resource and the CDMA region is short of network resource. In the middle of an on-line game, a user in the WLAN region decides to go outside to the CDMA region while continuing to play the game on a wireless device. The wireless device is continually serviced with degraded application quality on the screen, albeit there is a shortage of network resources in CDMA region. Therefore, the user can seamlessly enjoy the game on the wireless device.

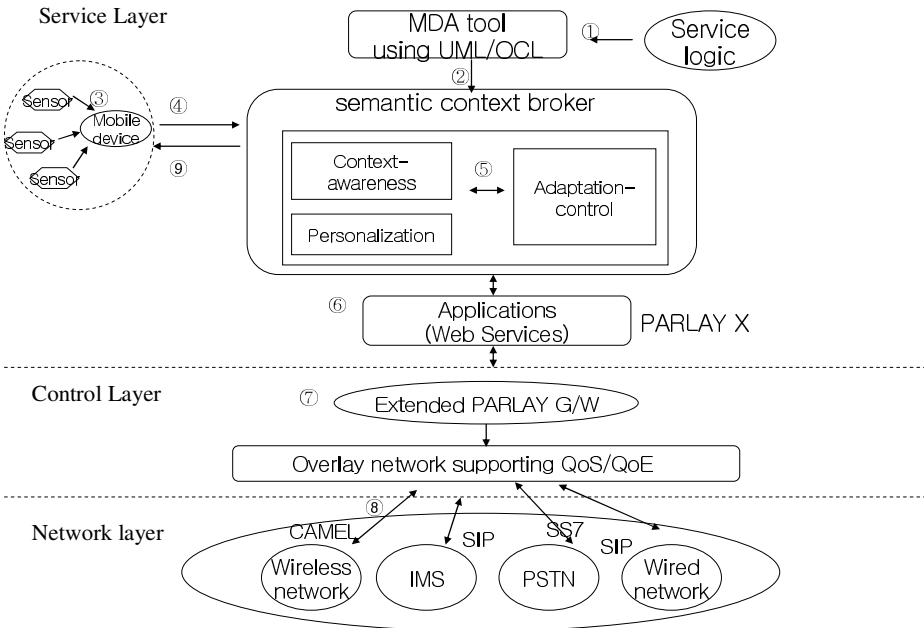


Fig. 1. The architecture of the extended PARLAY X for ACPS

We assume that semantic context broker can get the context such as location and the speed by GPS. The mobile devices including the intelligent agents can get the context such as weather, temperature from the sensors called motes. The mobile device informs the semantic context broker of the context.

Fig. 1 shows the overall architecture of ACPS which consists of the service layer, the control layer, and the network layer. This paper focuses on the service layer. The service layer includes Service Creation Environment (SCE) [10] and the semantic context broker. SCE is based on MDA technology and uses Unified Modeling Language (UML) / Object Constraint Language (OCL). SCE can specify the policy of

ISP or the network administrator. The semantic context broker is developed for the purpose of the extension of PARLAY X for adaptation, context-awareness, and personalization.. The adaptation function can get the context such as the policy from the ISP or the network administrator. The wireless sensor network called a mote can detect the contexts. The context-awareness function can get the context such as location, weather, and temperature from GPS or sensors. The personalization function can get the context such as the device type and the user's profile. The semantic context broker can analyze the context and decide the optimized network protocol. The semantic context broker informs the overlay network of the request of the optimized network protocol. The overlay network can support the protocol adaptation for QoS / QoE according to the request of the semantic context broker. Hence, the semantic context broker interprets the context for context-awareness and personalization according to the changing context and the user's preference and reconfigures the protocol for adaptation.

## 2.2 The Semantic Context Broker

The role of a MDA tool is to make a role of the SCE, which can obtain the service logic and the context constraints from the Internet Service Provider (ISP) or the network administrator. The service logic depends on a UML/OCL notation. We developed the semantic context broker for ACPS. The semantic context broker is for PARLAY X with QoS / QoE and to support context-awareness, personalization, and adaptation in the service layer as depicted in Fig. 2. The role of the semantic context broker is to obtain the context such as location and speed, to make an interpretation for context-awareness, and to re-composite each protocol for adaptation and personalization according to the context. The overlay network can support QoS / QoE according to the request of the semantic context broker in the control layer. ACPS uses MDA-based SCE with OCL and web service technology, whereas the existing PARLAY X uses XML-based web services technology. The signaling of PARLAY Gateway uses Mobile Application Part (MAP) in Wireless Network, SIP in IP Multimedia System (IMS), SS7 in Public Switched Telephone Network (PSTN), and SIP in wired network such as the Internet

In Fig.1, the operation of ACPS is as follows:

1. The MDA-based SCE defines the service logic and context-constraint.
2. The service logic and context-constraint using UML/OCL is transferred to the semantic context broker. The semantic context broker translates UML/OCL to PARLAY X.
3. Many sensors called motes inform the semantic context broker of the detected context information.
4. The mobile user's information such as user's preference and device type is transferred to the semantic context broker.
5. The semantic context broker converses the detected information into XML-based context information and reconfigures the service for adaptation according to the context.
6. The semantic context broker translates the XML-based context information into the XML-based PARLAY X.
7. PARLAY X is converted into PARLAY gateway.

8. The overlay network can support QoS / QoE according to the request of the semantic context broker.
9. Finally, ACPS provides the user with the customized network service with QoS / QoE .

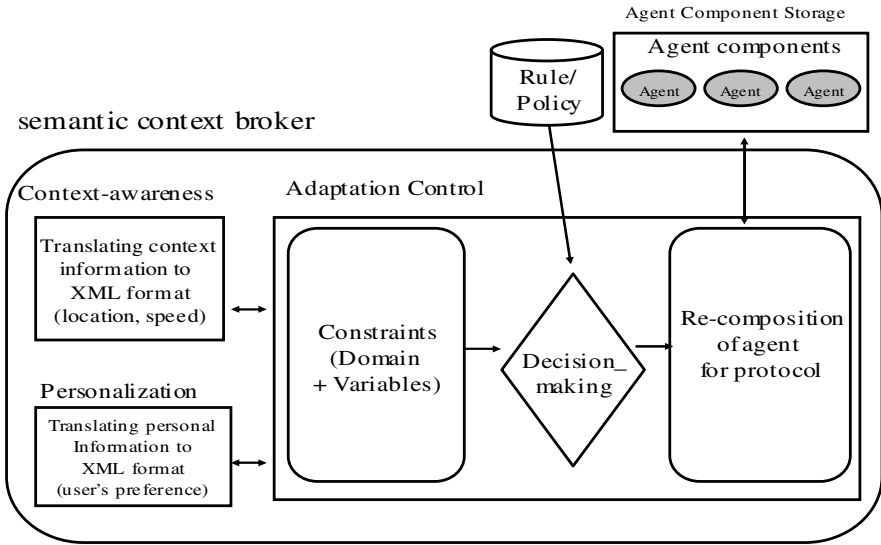


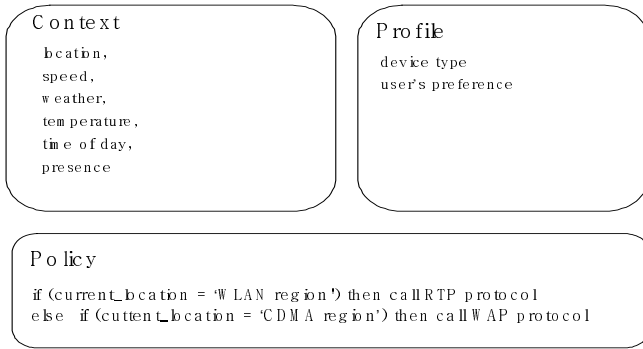
Fig. 2. Mechanism of the semantic context broker

Fig. 2 shows the mechanism of semantic context broker. The semantic context broker consists of *Context-awareness*, *Personalization*, and *Adaptation control*. *Context-awareness* has a role to interpret contexts that comes from mote or GPS. *Context-awareness* is to get the context information such as location and speed, translating the context information to XML format. *Personalization* has a role to process the user’s personal information such as user’s preference and the device type. *Personalization* is to get the personal information, translating the information to XML format. *Adaptation Control* is to reconfigure the protocol for adaptation according to the context. *Adaptation Control* is to re-composite the agent for protocol that can call network protocol module according the ISP’s rule and policy.

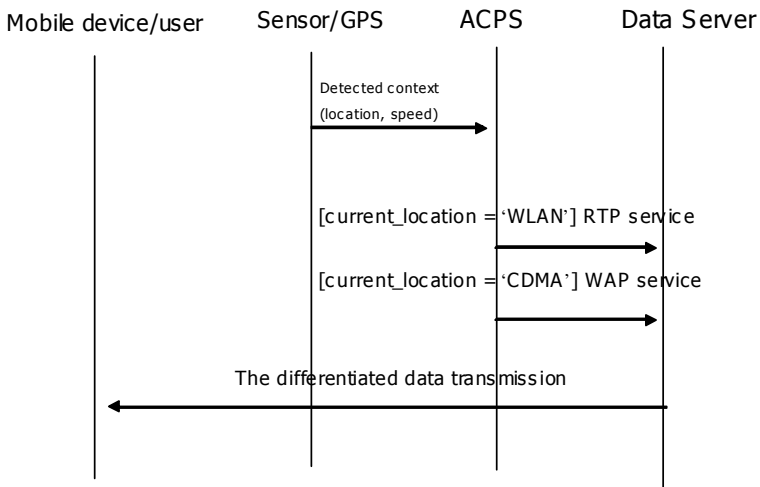
### 2.3 The Definition of Context, Profile, and Policy for ACPS

Fig. 3 shows the context, profile, and policy for ACPS. *Context* consists of *location*, *speed*, *weather*, *temperature*, *time of day*, *presence*, *device type*, and *user’s preference*. *Profile* consists of device type and user’s preference for personalization. The policy from the ISP or the network administrator is expressed by OCL. The example of the policy is as follows: *if (current\_location = ‘WLAN region’) then call RTP protocol else if (current\_location = ‘CDMA region’) then call WAP protocol* means to call a function for RTP protocol in the case that the current location is in a Wireless LAN region where resources of a network in the surroundings are enough,

and to call a function for WAP protocol in the case that the current location is in a CDMA region where the resources of network are scarce.



**Fig. 3.** Context, Profile, and Policy for ACPS



**Fig. 4.** Sequence diagram for ACPS

Fig. 4 shows the sequence diagram for ACPS. The sensor or GPS detects context information such as the location and speed and it informs context information of ACPS. ACPS can adaptively choose the optimized protocol, analyzing the context information and policy of the ISP. For instance, if the current location of a mobile device is in the WLAN region, users can get the high quality of service through Real Time Protocol (RTP), whereas if the current location of a mobile device is in the CDMA region, users can get the low quality of service through Wireless Application Protocol (WAP). Finally, the mobile user can get the differentiated network service.

### 3 Implementation of ACPS

The implementation of ACPS is based on Windows 2000 server, the PARLAY X SDK named GBox[12] of Appium company. The UML/OCL is converted into the XML-based web service because ACPS uses XML-based web services. We have three steps in the execution of ACPS. First, the policy using UML/OCL notation is defined by the ISP or the network administrator. In Fig.4, the example of policy is the expression using OCL. Second, the semantic context broker can get the context such as location, speed, weather, and temperature from the GPS or sensor, can get the context such as device type and user's preference from the mobile device, can translate UML/OCL into XML, can analyze the XML-based information, and can find the optimized network protocol. Third, the overlay network provides users with the network service with QoS / QoE according to the request of the semantic context broker. We use UML/OCL tool, PARLAY X SDK. ACPS includes the new defined PARLAY X API such as getContextAwareness(), getPersonalization, and adaptiveProtocolType(). We assume that there are WLAN region and CDMA region according to the horizontal (x) and vertical (y) axes of PARLAY simulator. ACPS can provide RTP protocol or WAP protocol according to the context information such as location.

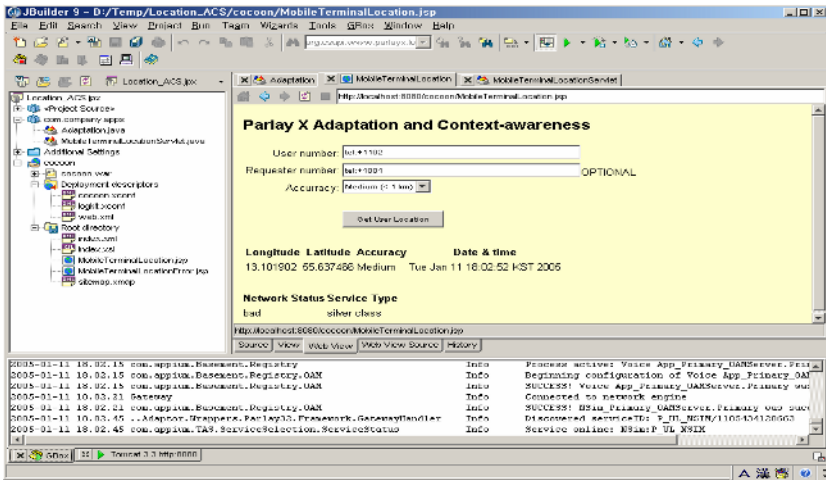
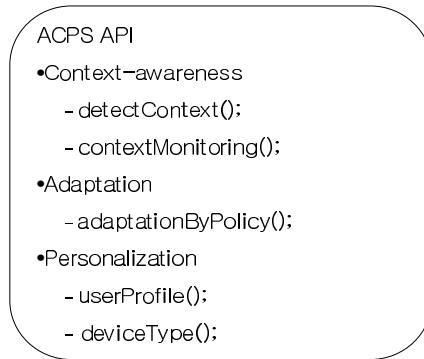


Fig. 5. The prototype for ACPS using GBox

Fig. 5 shows the prototype of the PARLAY X extension for ACPS using PARLAY X simulator called GBox. This prototype for ACPS can get the context such as the location. ACPS can decide to service the RTP protocol or the WAP protocol by the analysis of user's location. For instance, if the current location of wireless device is in the WLAN region, the ACPS provides the high quality service through RTP. If the current location of wireless device is in the CDMA region, the ACPS provides the low quality service through WAP.



**Fig. 6.** The defined PARLAY X API extension for ACPS

Figure 6 shows the defined PARLAY X API extension for ACPS including context-awareness, adaptation, and personalization. The defined API for context-awareness is named as *detectContext()* that can detect the context and *contextMonitoring()* that can monitor the context. The defined API for adaptation is named as *adaptationByPolicy()* that can support an adaptation by the ISP's policy. The defined API for personalization is named as *userProfile()* which can support user's preference and *deviceType()* which can detect the device type.

## 4 Comparison of the Features of the Existing PARLAY and ACPS

### 4.1 Comparison of Main Features

Table 1 shows the comparison of main features of the existing PARLAY X, WASP and ACPS. ACPS has more features, such as supporting the context-awareness, adaptation and personalization than the existing PARLAY X. ACPS considers MDA-based SCE using UML/OCL as the language for context-awareness, adaptation and personalization, whereas, the PARLAY X and WASP do not consider MDA-based SCE albeit they support the ad-hoc context-aware language. ACPS and WASP can support the context-awareness for location, speed, temperature, and weather, using the web service technology. ACPS can consider adaptation and personalization in the network. Conversely, PARLAY X and WASP do not consider the adaptation and personalization in the network.

**Table 1.** Comparison of main features

	PARLAY X	WASP	ACPS
Context-awareness	-	X	X
Adaptation	-	-	X
Personalization	-	-	X

### 4.2 Comparison of Performance

We evaluate performance using ns-2 simulator. There are four nodes for the performance evaluation in ns-2 like Fig. 4. The node 0 is for the mobile device. The node 1 is for GPS. The node 2 is for ACPS. The node 3 is for the data information server. The node 1 informs the node 2, which is ACPS, of the location of user, detecting it from the sensor or GPS. The node 2 is to re-composite the network protocol according to the network resource. We evaluate the packet size of data that is sent to the user. We define the ChangingContext() method using C++ programming language in ns-2 for evaluation in case that the context is changed.

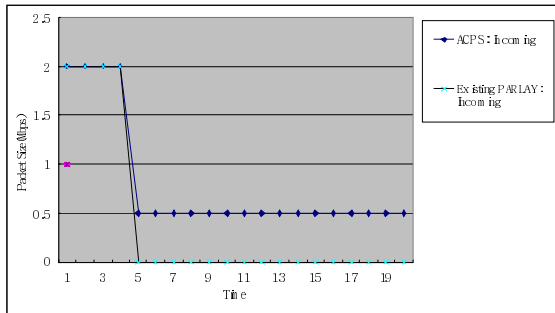


Fig. 7. Comparison of performance

Fig. 7 shows the comparison of performance on the feature of adaptation and personalization between the existing PARLAY X and ACPS. The existing PARLAY is stopped in case that the current location in WLAN is changed in CDMA region. Conversely, ACPS can keep the service because the RTP protocol service is changed to the WAP protocol service in case that the current location in WLAN is changed in CDMA region. This is attributed to the fact that ACPS supports adaptation and personalization, the existing PARLAY does not have adaptation and personalization functionality.

## 5 Conclusion and Future Works

This paper suggests the extended PARLAY X and the open API extension to support context-awareness, adaptation and personalization for QoS / QoE. We believe that ACPS addresses new service mechanism on delivery network platform to support more QoS / QoE on the network than the existing PARLAY X. We expect ACPS to comply with the industry standard such as PARLAY. Our future work will involve more studies applying open API extension on the standard on the reconfigurable Software Defined Radio (SDR) equipment of Wireless World Research Forum (WWRF) [13] for 4G.



## References

- [1] PARLAY home page : [www.parlay.org](http://www.parlay.org).
- [2] O. Kath, T. Magedanz, R. Wechselberger: "MDA-based Service Creation for OSA/Parlay within 3G beyond Environments" First European Workshop on Model Driven Architecture with Emphasis on Industrial Application, March 17-18, 2004. University of Twente, Enschede, The Netherlands.
- [3] IKV++ Technologies AG: The medini tool chain, available at <http://www.ikv.de/pdf/mediniWhitePaper.pdf>.
- [4] S. Pokraev et. al., "Context-aware Service," Technical Report WASP/D2.3, Telematica Instituut, Ericsson, November 2003, available at <https://doc.telin.nl/dscgi/ds.py/Get/File-27859/>.
- [5] C. Efstratiou, K. Cheverst, N. Davies and A. Friday, "An architecture for the effective support of adaptive context-aware applications," Proceedings of 2nd International Conference in Mobile Data Management (MDM'01), Hong Kong, Springer, Vol. Lecture Notes in Computer Science Volume 1987, pp. 15-26, January, 2001.
- [6] Barbir et al., "A Framework for Service Personalization", draft-barbir-opes-fsp-03.txt, work in progress, March 2003.
- [7] Timothy M. O'Neil, "Quality of Experience and Quality of Service, For IP video conferencing," White paper by Poly com, <http://www.h323forum.org/papers/polycom/QualityOfExperience&ServiceForIPVideo.pdf>.
- [8] Harry Chen, Tim Finin, Anupam Joshi, "Semantic Web in the Context Broker Architecture", In Proceedings of PerCom 2004, Orlando FL., March 2004.
- [9] J. Wing., and Kleppe, A., "OCL : The Constraint Language of the UML," JOOP, May, 1999.
- [10] Bernhard Stenffen, Tiziana Margaria, Andreas Claben, Volker Bruan, "A Constraint-Oriented Service Creation Environment," 2<sup>nd</sup> International Conference on Practical Application of Constraint Technology, London (UK), April 1996.
- [11] L. Girod, J. Elson, A. Cerpa, T. Stathopoulos, N. Ramanathan, D. Estrin, "EmStar: a Software Environment for Developing and Deploying Wireless Sensor Networks," in the Proceedings of USENIX General Track 2004.
- [12] GBox home page - PARLAY X Service Creation Environment, <http://www.appium.com>.
- [13] Wireless World Research Forum home page : <http://www.wireless-world-research.org>.

# A Context-Aware System for Smart Home Applications

Wen-Yang Wang<sup>1</sup>, Chih-Chieh Chuang<sup>1</sup>, Yu-Sheng Lai<sup>1</sup>, and Ying-Hong Wang<sup>2</sup>

<sup>1</sup> Computer and Communications Research Laboratories,  
Industrial Technology and Research Institute, Tainan City, Taiwan  
{wwj, twjack, laiys}@itri.org.tw

<sup>2</sup> Department of Computer Science and Information Engineering, Tamkang University,  
Taipei County, Taiwan  
inhon@mail.tku.edu.tw

**Abstract.** Context-awareness is an important part for ubiquitous computing. Many applications of ubiquitous computing have to access some related contexts in order to provide the right services at the right time and the right places. However, there are some challenges for applications in ubiquitous computing, especially for those in smart home. These challenges make the design of smart home applications much more difficult than other applications. Therefore, we propose a context-aware system, CASSHA (Context-Aware System for Smart Home Applications), which is designed for smart home applications. CASSHA consists of the components for processing, representation, provision, and coordination, and is able to provide required contexts for smart home applications without conflicts. The system overcomes most of the technical challenges for smart home applications, and satisfies the requirements for these applications.

## 1 Introduction

In ubiquitous computing, most of the applications may need to collect several different contexts. The example of contexts may include user identification, positioning and tracking, user's activities, as well as time, temperature, et al. These contexts are required for the applications in the field of ubiquitous computing, especially for those in smart home. To make a "smart" decision, the applications may need much information of the user and the surroundings. Take the tour guide application as the example, the application needs to know the user's current location and his destination, and then points out a path for the user. The path may be merely the shortest path, or may be a less crowded one according to other users' activities and events in the environments. In other words, the later one will be the path that will have less people and traffics when the user passes through. In order to compute the less crowded path, the application needs to be aware of more contexts other than the source and destination of the current user. Thus, context-awareness is needed for applications of ubiquitous computing.

In order to collect more contexts, different types of sensors will be needed. For examples, thermometers measure the temperatures, while locators show the locations of users. However, the data generated by these sensors are simply raw data and cannot

be easily used by applications. Besides, some contexts cannot be retrieved directly from single type of sensors, and may only be retrieved after several analyses. An example is user activity. If a user is sitting on the sofa in the living room, and the television is turned on, then it is possible that this user is watching TV. In this case, both user's location and the status of the TV are required to decide the user activity of watching TV. If more information is retrieved, including the user's preference, content of TV program, and user's viewpoints, then the analyzed result will be more precise. As we can see, to determine user activities precisely is too complex for a single application. Therefore, a context-aware system for providing and analyzing context is needed for ubiquitous applications.

For smart home applications, there are much more difficulties on designing the system. Infrastructure at home may not be sufficient for supporting new applications, devices may not be bought at the same time, no system administrator is available, and user activities are more difficult to predict at home than in other places. Besides, there are other social implications and marketing issues that also increase the difficulties for application in smart home. Therefore, designing context-aware systems for smart home applications is more difficult than systems for other applications.

In this paper, we summarize the challenges of smart home applications from previous researches, and design CASSHA as a context-aware system that can overcome, or at least mitigate, these challenges. Our design uses a layered approach, including Context Processing, Context Representation, Context Provision, and Application Interface. In each layer, functionalities are also defined in order to overcome or mitigate the challenges of smart home applications.

The rest of paper is organized as follow. Section 2 first addresses the challenges of smart home applications. And in section 3, previous works of context-aware systems are discussed. Afterwards, CASSHA is presented in section 4. Finally, section 5 concludes our work and discusses issues for future researches.

## 2 Challenges

As people can expect, smart home becomes the trend of future homes. Smart home is able to integrate technologies of digital living and provides a comfortable, secure, and convenient living style. However, smart home also introduces challenges for designing applications in this environment. Smart home applications differ from others in many aspects. Some researches [5] have been presented to address these issues and the impacts of smart home applications. These challenges includes:

1. **Seamless Platform.** There are various types of devices at home: multimedia servers, intelligent monitors, home appliances, as well as sensors and controllers. It is unpredictable that when and where these devices will be installed. In order to connect and maintain interoperability of these "accidental" devices, not only the standards of network transmission should be considered, but also the construction of a seamless platform is needed. With the seamless platform, various smart home applications can thus cooperate with each other.
2. **Mechanism for Open Service Management.** There will be many different smart home applications in the future, and the number of these applications will grow up as the time goes by. On the one hand, new applications should be able

to add into the smart home when they are presented. On the other hands, some old applications may be updated with more powerful functionalities. Therefore, there must be a mechanism for managing smart home applications, in order to dynamically add new applications and keep upgrading the old ones.

3. **Convenience and Reliable with No System Administrator.** Home users do not have enough knowledge and experiences of ubiquitous computing technologies, nor do they have a dedicated administrator to manage and maintain the applications at home. However, smart home applications should still be able to execute under these circumstances. Therefore, these applications must be convenience and reliable, even when no system administrator is presented.
4. **Inference under Various Requirements and Situations.** Home users may use smart home applications under different requirements and situations. Moreover, some away-from-home activities may also impact these applications. Therefore, applications must fulfill various requirements and situations, and provide expected results by inferences.
5. **User-oriented Designs.** There will be plenty of applications designed for different scenarios and use cases in smart home. However, these applications should be designed based on users' requirements, and should provide a convenient interface if user interactions are needed.

As we can see, the challenges of smart home applications are more difficult than other applications. In order to reduce the difficulties for designing smart home applications, a context-aware system that manipulates these challenges is needed; hence, the design of this system is much more difficult than other systems.

### 3 Related Work

Previous researches have introduced different approaches of the context-aware systems for different applications. Though the context-aware system for smart home applications is not a new research topic, former researches cannot fulfill all the requirements of these applications.

Lee and Chung proposed system architecture for context-aware home applications [9]. They classified home applications into several categories, and provided four scenarios for media, healthcare, control, and management applications respectively. After that, they analyzed the required technologies for each scenario, and proposed a structure of home application server that can integrate these technologies. However, this home application server saves only raw data in its database, and does not have a well-defined representation for contexts. These will provide less reusability of contexts and lead to the difficulties of making inferences.

Another context-aware system for home applications can be found in [10]. Universal Home Network Middleware (UHNM) uses Adaptors to achieve network interoperability, uses several managers to take care of applications, and uses Messaging Layer for communication between managers. It is designed to provide zero-configuration, high-level abstraction, context-awareness, and adaptation. However, UHNM does not have well manipulation of contexts; it only utilizes the data from sensors and cannot perform inferences from sensed data at all. Besides, UHNM also lacks of a database for recording the events and the sensed data.

Authors in [2] also proposed a context-aware middleware for controlling home appliances. The middleware is based on OSGi (Open Service Gateway Initiative) and uses UIML (User Interface Markup Language) to define user interfaces. This middleware focuses on the learning and prediction of user preferences, and uses neural networks to do the learning and prediction. However, only user preferences can be provided by the middleware, which may not be sufficient for all the smart home applications. Besides, due to the nature of neural networks, the training data should be large enough in order to make good predictions—this may cause inconvenience for users at home, and needs an experienced user to master the training process.

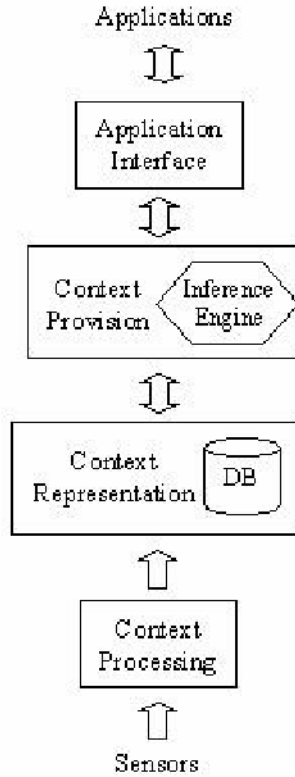
Still some related work takes different approaches and focus on other aspects. In [7], a Service-Oriented Context-Aware Middleware (SOCAM) has been proposed. The key feature of SOCAM is its ability to reason about various contexts. And in [8], an agent-based approach is used. Mobile agents travel between devices to collect data and perform activities of the applications. Both these two approaches do not have the mechanism to coordinate applications, and may lead to unpredicted results and reduce the reliability of the systems. Other systems such as [3], [4], and [6], are not designed for smart home applications. Most of these systems require infrastructure and administrators, which are not appropriate for smart home applications.

## **4 Context-Aware System for Smart Home Applications (CASSHA)**

According to the challenges mentioned in section 2, we propose CASSHA for smart home applications. CASSHA is designed to overcome or mitigate these challenges of smart home applications. It is a seamless platform for the interoperability of various devices at home, provides an interface for applications to be managed, no administrators are needed, inferences can be made under various requirements and situations with its inference engine, and it is designed for usage of smart home applications. CASSHA includes four components in order to achieve these goals. These components are shown in Fig. 1, namely Context Processing, Context Representation, Context Provision, and Application Interface. We will discuss these components respectively in the followings.

### **4.1 Context Processing**

Context Processing does the job to collect the raw data from sensors and to interpret contexts thereafter. It collects the raw data from all sensors at home. In order to achieve that, it bridges the physical networks, identifies the type of sensors, and reads the measurements from them. As mentioned before, these sensors may be bought at different time, and may be installed or replaced at some other time. Once a sensor is installed, Context Processing will be aware of these sensors through additional bridges and adapters. Then it identifies the type and the location of these sensors through some service discovery mechanisms (such as UPnP [12]) and localization methods. When a sensor is removed or unable to response, Context Processing then marks the sensor as unavailable. By these means, Context Processing is able to adapt the present and absent of the sensors.



**Fig. 1.** The proposed context-aware system for smart home applications

After collecting the raw data from sensors, Context Processing then interprets contexts from these raw data. It will interpret the raw data according to the sensors’ types, characteristics, and their locations. For example, the data read from the thermometer in the living room will be considered as the room temperature of the living room. In addition to the interpretations of simple contexts, Context Processing will also combine the related contexts. Contexts sensed in the same place and at the same time will be combined together, as they may be highly related to each other.

#### 4.2 Context Representation

Context Representation normalizes the contexts collected from Context Processing and saves the normalized contexts into the database in order to query and analyze. It plays the role for improving the reusability of collected context. When receiving the context from Context Processing, Context Representation manipulates the contexts and translates them into a normalized representation. The representation may be ontology-based or in other forms [1, 11].

After normalizing, these contexts will be saved into a database. With this database, contexts can be reused for any required applications. Once contexts are saved, the

applications can query contexts through Context Provision at any time, and Context Provision can also analyze these contexts in order to produce other high-level contexts. Take user tracing for example. In order to obtain the path that a user moves, previous contexts of the user's location will be needed and these contexts can be accessed from the database to enhance reusability. Another example is that for the learning mechanisms in Context Provision, former status of contexts may be needed, and these contexts can also be accessed directly from this database.

### **4.3 Context Provision**

Context Provision provides the contexts for smart home applications. When an application requires some contexts, Context Provision acts as an interface to query contexts from the database in Context Representation. If the needed contexts are simple contexts and can be found in the database, Context Provision will soon return these contexts to applications. However, if high-level contexts such as user activities are needed, Context Provision will have to analyze the contexts in the database and provide the results to applications.

In order to perform the analysis, Context Provision has an inference engine to do the job. The inference engine takes several contexts, as well as user profiles and environment layouts into consideration. Therefore, the results of high-level contexts can be more meaningful than the contexts simply produced by Context Processing. In addition, with well-designed user profiles and a good learning mechanism, the user activities at home can become more noticeable, and the difficulties of analysis can also be reduced as well. For example, high-level contexts such as user activities can be obtained through the inference engine. Although some of early analyses of user activities may lead to unexpected results, with the learning mechanism and user profiles, the accuracy will be enhanced after several trials.

### **4.4 Application Interface**

The main function of Application Interface is to provide an interface for smart home applications to access the system. This interface hides the lower layer details from applications. Applications only have to know what contexts they required, and do not have to know how to get these contexts from which sensors.

In addition to provide the interface, Application Interface has another important function as well—to resolve the conflict between applications. In normal situation, Application Interface is simply an interface for applications to access the system. However, when conflicts occur between two or more applications, Application Interface has to coordinate the conflicted applications. These conflicts may result from two or more applications designed for different situations. For example, an energy reservation application will turn off the air conditioner if no one is at home, but another application may turn on the air conditioner when users are arriving home. In this case, Application Interface must coordinate these two applications according to when will users arrive home. When the number of home applications grows up, the occurrences of such conflicts will become more frequently. Therefore, Application Interface plays an important role to resolve these conflicts.

## 4.5 Discussion

With CASSSHA, we can overcome or mitigate the technical challenges of smart home applications. CASSHA is indeed a seamless platform for smart home applications. The interoperability is achieved by Context Processing. Once devices and sensors are added, Context Processing will take care of the installation and configuration of these devices and sensors, and make use of them. Context Processing collects different type of sensors and shields the heterogeneity from upper layers. Besides, it can merge different physical networks at home with some additional adapters or bridges for each network.

As for the challenges of application management, Application Interface does a good job. The interface is provided not only to the applications at home, but can also be provided to service providers under user's permission. Therefore, service providers are able to use the interface to see what applications can be added in the smart home. Besides, old applications can be updated through the same mechanism.

When it comes to the challenge of convenience and reliable with no system administrator, all of these four components contribute to overcome this challenge. Context Processing provides the interoperability of devices and sensors, which may ease the installation progress for home users. Context Provision learns and analyzes user preferences, while Application Interface resolves the conflicts of applications. With these two components, administration is only needed when experienced users want to tuned the system for more detailed settings. And reliability is achieved with the assists of Application Interface and the database in Context Representation. Due to Application Interface resolves the conflicts of applications, thus the possibility of system failure will be reduced. However, if a failure still occurs, the database in Context Representation provides capability for system to recover.

The next challenge is about the inference. Obviously, Context Provision is designed to mitigate this challenge. It provides the required contexts to applications that are designed under various requirements and situations. Simple contexts can be retrieved from database directly, while the inference engine can extract high-level contexts. With additional information, well-designed user profiles, and a good learning mechanism, the accuracy of the analysis will be high enough for smart home applications.

The last challenge mentioned is the user-oriented design. The solution of this challenge may refer to Application Interface. Although it does not provide any user interface, the provided interface for smart home applications are well enough for these applications to design the user-oriented user interfaces.

To sum up, CASSHA overcomes the challenges of smart home applications with the functionalities of its four components: Context Processing, Context Representation, Context Provision, and Application Interface. Challenges of smart home applications are overcome by either a single component or by several components among four of them. These four components are well bound to each other and satisfy the requirements of smart home applications.

## 5 Conclusion and Future Work

In this paper, we have proposed CASSHA, a context-aware system for smart home applications, to overcome the challenges of smart home applications. CASSHA



consists of four components, namely Context Processing, Context Representation, Context Provision, and Application Interface. With the functionalities of these components, the challenges for smart home applications can be overcome or mitigated.

However, there are still some other challenges for smart home applications. One of these challenges is the impacts of ethical, legal, and social implications. When smart home applications become popular in human's living, these new technologies will lead to some non-technical issues. The violation of privacy, the influence of living styles, the transformation of social structure, and so forth. These impacts may be hardly predicted, and can only be analyzed through a thorough research on behaviors of human beings.

## References

1. Chen, H., Finin, T., Joshi, A.: An Ontology for Context-Aware Pervasive Computing Environments. *Special Issue on Ontologies for Distributed Systems, Knowledge Engineering Review*, volume 18, issue 3, pages 197-207, May 2004.
2. Choi, J., Shin, D., Shin, D.: Research and Implementation of the Context-Aware Middleware for Controlling Home Appliances. *IEEE Transactions on Consumer Electronics*, volume 51, issue 1, pages 301-306, February 2005.
3. Costa, P. D.: Towards a Services Platform for Context-Aware Applications. *Thesis for a Master of Science degree in Telematics from the University of Twente, Enschede, The Netherlands*, August 2003.
4. Covington, M. J., Fogla, P., Zhan, Z., Ahamad, M.: A Context-Aware Security Architecture for Emerging Applications. *Proceedings of the 18<sup>th</sup> Annual Computer Security Applications Conference (ACSAC 2002)*, pages 249-258, December 2002.
5. Edwards, W. K., Grinter, R. E.: At Home with Ubiquitous Computing: Seven Challenges. *Proceedings of the Third International Conference on Ubiquitous Computing (UBICOMP 2001)*, pages 256-272, September 2001.
6. Efstratiou, C., Cheverst, K., Davies, N., Friday, A.: An Architecture for the Effective Support of Adaptive Context-Aware Applications. *Proceedings of the Second International Conference on Mobile Data Management (MDM'01)*, pages 15-26, January 2001.
7. Gu, T., Pung, H. K., Zhang, D. Q.: Toward an OSGi-Based Infrastructure for Context-Aware Applications. *IEEE Pervasive Computing*, volume 3, issue 4, pages 66-74, October-December 2004.
8. Hattori, M., Cho, K., Ohsuga, A., Isshiki, M., Honiden, S.: Context-Aware Agent Platform in Ubiquitous Environments and its Verification Tests. *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications (PerCom'03)*, pages 547-552, March 2003.
9. Lee, S. H., Chung, T. C.: System Architecture for Context-Aware Home Application. *Proceedings of the Second Workshop on Software Technologies for Future Embedded and Ubiquitous Systems (WSTFEUS'04)*, pages 149-153, May 2004.
10. Moon, K. D., Lee, Y. H., Kim, C. K.: Context-Aware and Adaptive Universal Home Network Middleware for Pervasive Digital Home Environment. *IEEE Consumer Communications and Networking Conference 2004 (CCNC 2004)*, pages 721-723, January 2004.
11. Strang, T., Linnhoff-Popien, C.: A Context Modeling Survey. *First International Workshop on Advanced Context Modeling, Reasoning, and Management (UbiComp 2004)*, September 2004.
12. UPnP<sup>TM</sup> Device Architecture. *UPnP Forum*, June 2000.

# Human Position/Height Detection Using Analog Type Pyroelectric Sensors

Shinya Okuda, Shigeo Kaneda, and Hirohide Haga

Graduate School of Engineering, Doshisha University,  
1-3 Tatara-Miyakodani, Kyotanabe City 610-0321, Japan  
tam@ishss10.doshisha.ac.jp

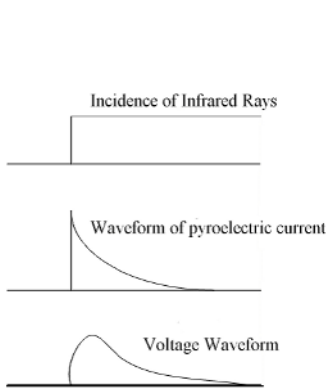
**Abstract.** Pyroelectric sensors can detect changes of infrared rays. Usually, typical pyroelectric sensors are the digital output type, used for lighting switches and security systems. We can acquire, however, limited information from such digital on or off signals. We can also acquire a wealth of sensor data by using the analog type. Thus, this paper proposes a new approach to human position detection that discriminates human height (adults or children) by using analog type pyroelectric sensors. The proposed method employs Fast Fourier Transform (FFT) to calculate human position and analyzes the spectrum distribution to discriminate between adults or children. We built an experimental room 2.5 meters square and 2.5 meters high. Analog type sensors were installed at intervals of 0.8 meters in a grid shape. The proposed position detection method can calculate human position even if two persons are in the same room. Our height detection method that discriminates between adults and children is almost 90% accurate. Future research targets are improvements in accuracy and the development of an application system using these sensors.

## 1 Introduction

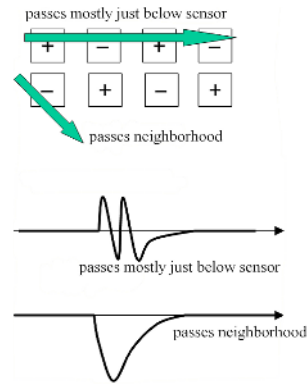
Pyroelectric sensors detect the feeble far-infrared rays emitted by the human body. They are usually used in lighting switches and security sensors. For instance, they are installed in rest room ceilings and passageways to save energy. These sensors have the following features: 1) can be used in a completely dark room without light; 2) no invasion of privacy problems; 3) nothing to be equipped for human body.

We arranged these sensors in the shape of a lattice on the ceiling and determined the position and height detection techniques from the data when examinees passed through their bottom. However, typical digital-type far-infrared sensors are immediately activated if a detected object enters detection range and this state continues for a fixed time. For this reason, the information from sensors is either on or off. We cannot detect when examinees passed directly under sensors.

Thus, we decided to treat the data of sensors as analog signals. In addition, we propose timing detection techniques when passing the sensors and for height differences between adults and children to consider the input of context-aware services corresponding to adults and children.



**Fig. 1.** Theoretical Waveforms



**Fig. 2.** Polar Characteristics of Detection Area and Output Waveforms

The remainder of this paper is organized as follows. In Section 2, we explain the waveforms of analog type pyroelectric sensors and in Section 3 propose detection techniques. In Section 4, we evaluate our experiments. Conclusions are drawn in Section 5.

## 2 Output Waveforms of Pyroelectric Sensors

Conventional position and height detection techniques use image processing, RFID (Radio Frequency Identification), ultra-sonic wave and etc. They can acquire fixed accuracy, but it is difficult to use them in everyday life. Image processing requires lighting system which is not available for sleeping time. RFID tag is far from handsfree. Ultra-sonic sensor requires endless ultra-sonic power emission. Since these demerits are practically serious obstacles, we used a different device and approach.

First, we focus the output waveforms of having one pyroelectric element. With a pyroelectric element, if the element's temperature rises by infrared rays, polarization will arise. Infrared rays are detected by locating the current produced when it is neutralized. Then the current is amplified with a DC amplifier with very high input impedance. Figure 1 shows the theoretical output signal waveform.

As shown in the upper part of Fig. 1, the amount of infrared rays that conduct incidence to an element increases. The current flowing from the pyroelectric elements is shown in the middle part of Fig. 1. Finally, the outputted signal is shown at the bottom. These sensors detect only changes in the amount of infrared rays; they can detect human movement, but not when the movement stops. The rising edge of output voltage is steep, and the output voltage tail is comparatively loose.

Below we consider a passing object as a shape of dot. Looking at the top of Fig. 2, "passes mostly just below a sensor" shows that a passing object goes through the plus and minus areas in sequence. First, the voltage swings to the

plus direction, and then the passing object immediately moves to the next minus area. At this time such a phenomenon as the “signal will go up to positive, but flicks off in the opposite direction” occurs. By the same token, the infrared ray source moves to the next positive area and swings over into the positive direction again.

Finally, in cases of “passes mostly just below the sensor,” the signal’s characteristic is comparatively small amplitude, and the cycle is also short, except the tail, until the electric charge in an element is stabilized. (See the bottom of Fig. 2). The cycle depends on the source of the infrared passage speed. Frequency becomes high, so the movement of the object is quick, and amplitude becomes small. On the contrary, when it passes so that one domain may be grazed (passes the neighborhood in Fig. 2), the frequency ingredient is low, and amplitude increases. Consideration is summarized to below.

- Although the tail portion waveform has a long cycle and large amplitude, the object has already passed underneath a sensor by then.
- A high frequency ingredient does not appear if the infrared ray source does not pass through the center of the detection area. Therefore, if a high frequency ingredient is taken out, it is possible to capture the timing at which the infrared ray source has passed the detection range of a sensor.

### 3 Proposal Techniques

In this section, we describe the position detection and height difference detection techniques that used pyroelectric sensors. Moreover, these techniques depend on passage speed for output signals. Since, we compute passage speed between sensors, how to add this to the parameters of height difference detection is described.

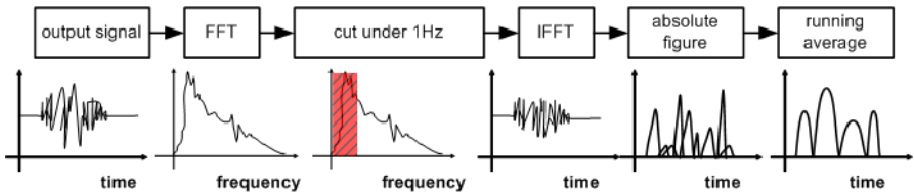
#### 3.1 Position Detection Technique

The sensors output signals after the infrared ray source has passed the detection area and until the electric charge in the elements are stabilized, as in Fig. 2. Since the portions of the last signals are low frequency waves, it is difficult to obtain the actual time of passing from raw waveforms. Furthermore, since the output value of a signal is also changed by passage speed, the temperature of the measurement environment, etc., the output maximum cannot simply be the recent side of a sensor. Time series detection of passing order is also difficult because part of the detection domain overlaps the contiguity sensors and output is mixed.

Since the waves of a low frequency ingredient are contained in the signals, the above problem is produced. Therefore, we developed the following procedures to detect human position. The processing outline is shown in Fig. 3.

#### [Position Detection Technique Procedure]

- STEP 1: Waveforms transform a time domain into a frequency domain using Fast Fourier Transform (FFT).



**Fig. 3.** Processing Procedure of Position Detection

- STEP 2: Cut the comparatively low frequency ingredient, which is set under 1 Hz in this evaluation.
- STEP 3: Transform a frequency domain into a time domain using Inverse Fourier Transform (IFFT)
- STEP 4: Calculate the running average every 150 data.
- STEP 5: Peaks are assessed as points of passing under the sensors.

The output signals of pyroelectric sensors are assumed to be unsaturated by this technique. Therefore, when there are two or more human beings, each effect will probably show up as output as an effect of liner superposition. On the contrary, such a frequency ingredient as “passes mostly just below sensors” appears. The span that appeared in high frequency is not passing. The infrared ray source “exists” in the detection domain of the sensors in that span. Even if the object has passed some areas at the edge of the entire detection domain, a peak appears in the timing. Moreover, the above argument is applicable in cases where an object passes through the domain of “+” and “-” in alternate shifts. Generally, it may pass through a domain aslant. In this case, the possibility that a complicated frequency ingredient will appear cannot be denied.

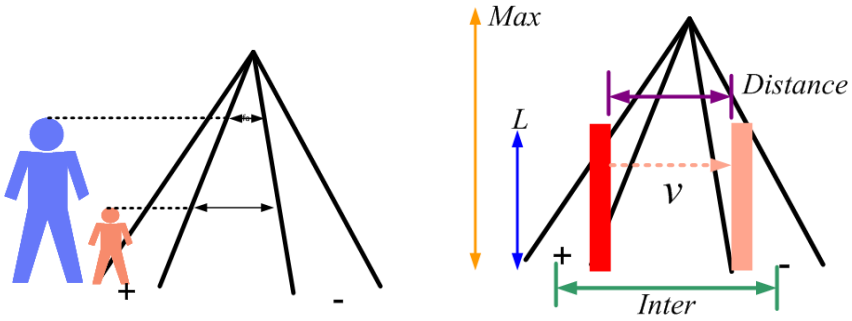
### 3.2 Height Detection Technique

In this section, we describe a height difference detection technique that measures the height differences of humans who enter detection areas and discriminates the height differences of children from adults (more than 170 cm and less than 100 cm) considering the input as context-aware.

First, the relation between height and detected frequency is considered. A sensor has a detection range of two or more pluses and minus. Detection range becomes large when passing through the place distant from the sensor. When tall, the portion with a narrow detection area interval is passed, and when low, the portion into which the interval has spread is passed. (Refer to Fig. 4) As a result, when tall, the wave that contains a higher frequency ingredient than low cases is outputted.

The above point is considered theoretically. As shown in Fig. 5, sensors are installed in the ceiling and at points in the floor to establish the following values:

- $Max[m]$ : distance from a ceiling to floor
- $Inter[m]$ : distance between centers of the detection domain in the  $Max$



**Fig. 4.** Height Difference and Passage **Fig. 5.** Theoretic Model of Detection Area Position

- $L$ [m] : height of the human being detected
- $v$ [m/s] : passage of time

At this time, we assume that infrared source is human’s head. The distance ( $Distance$  [m]) between the centers of the detection range of the height of the head is shown by Formula 1.

$$Distance = \frac{Inter \times (Max - L)}{Max} \text{ [m]} \tag{1}$$

One wavelength approximately corresponds to twice the distance of the centers of the detection range.

$$F = \frac{v}{\lambda} = \frac{v}{2 \times Distance} \text{ [Hz]} \tag{2}$$

Therefore, from the specification of sensor, when  $Max$  is 2.5 m,  $Inter$  is about 0.4 m. Frequency is  $F = 3.1$  Hz, if the height is  $L = 1.7$  m when a man’s walking speed is set to  $v = 0.8m/s$ . On the other hand, if there is only  $L = 0.8$  m height, frequency  $F = 0.92$  Hz will become the highest frequency. The height difference turns into a frequency difference.

The power spectrum of frequency is compared with cases of being tall and short (expressed as “adult” and a “child” below for convenience.) In the former, an amplitude of 1 ~ 3 Hz is strong, and in the latter, ingredients of over 1 Hz are seldom included and the amplitude is considered small. Furthermore, amplitude is more often low for adults, on the whole, than for children’s cases.

Thus, when signal waveform is transformed into a frequency domain using FFT, height differences notably appear as the difference of included frequency ingredient. The domain of characteristic frequency is established on the basis of the highest frequency  $F$  Hz (Formula 2).

As mentioned above, the discrimination of adults and children is possible by comparing the power spectrum of a characteristic frequency domain. Actually, it processes stepping on the following procedure.

### [Height Detection Technique Procedure]

- STEP 1: Process FFT, about the data which human passed under the three sensors.
- STEP 2: Calculate the sum total value of the power in the three sensors.
- STEP 3: A low frequency domain is set to *Field I* and a high frequency domain is set to *Field II*, as sum total values.
- STEP 4: Calculate the average value of the power in *Field I* and *Field II*.
- STEP 5: Calculate the rate of *Field II* on the basis of *Field I*. This value is set to the *Ratio*.
- STEP 6: Judge as an adult or child if the *Ratio* is beyond or below the threshold.

Output waveform has considerable variation according to the measurement environment. So, it is not commonly understood that the absolute value of amplitude applies to appraisal standards. To solve this problem, we adopt *Ratio* as a relative value, which is calculated as follows.

$$Ratio = \frac{average(Field II)}{average(Field I)} \quad (3)$$

### 3.3 Speed Detection Technique

The height difference detection technique mentioned above calculated a man's passage speed as about 0.8 m/s and computed the *Ratio* value. Although 0.8 m/s was said to be general walking speed, we also considered the detection of slower speeds ( $v = 0.5$  m/s). Since the speed is slowed down, the range in which the features of an adult and a child appear in a power spectrum is basically shifted to a low frequency domain. Concrete passage speed calculation is performed in the following order.

#### [Speed Detection Technique Procedure]

- STEP 1: The time when the peak value of the beginning at the time of sensor domain penetration appears is detected in three sensors.
- STEP 2: Calculate the time passage between the first and second sensors and between the second and third.
- STEP 3: Passage speed is calculated from installation distance and time passage of sensors.
- STEP 4: The average value of the speed of two passages is  $v$  m/s.

The height difference judging technique in which passage speed was also considered is described as follows. If  $v$  m/s is beyond the threshold, it is set to "normal." If not, it is set to "slow." Then an applicable *Field* is set, and height difference is judged.

## 4 Evaluations by Experiments

### 4.1 Experimental Equipment

Figure 6 shows a sensor circuit that we built as a prototype. We installed nine circuits in the shape of a lattice at intervals of 0.8 m and heights of 2.5 m. All equipment is shown in Fig. 7. The sensor portion is analog output and spot detection type NaPiOn (Model Number: AMN23112) by Matsushita Electric Works, Ltd.

Normal NaPiOn sensors are too sensitive for correct measurements since the signal is saturated. It's difficult to analyze frequency from saturated signals. To attenuate infrared rays, a polyethylene board was placed in front of a sensor. When installation interval was 0.8 m, overlap to the next sensor was large. To narrow the detection area, tape was stuck to the lens. Ideally, it should become the detection range shown in Fig. 8.

Sensor output is transmitted to computer by balanced transmission through a multicore shielded cable to avoid external noise influence at the time of signal transmission. An AD conversion board (from Interface Corporation, Japan) receives from the PC side. Sampling rate is 100 Hz.

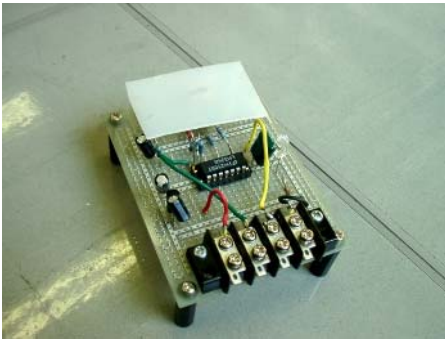


Fig. 6. Circuit



Fig. 7. Prototype System

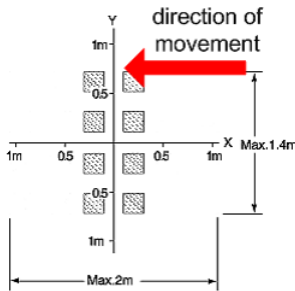


Fig. 8. Detection Range after Adjustment

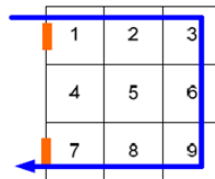


Fig. 9. Route for an Experiment



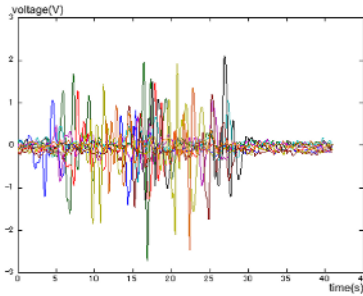


Fig. 10. Output Waveforms

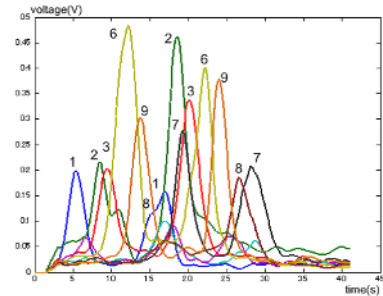


Fig. 11. Processed Waveforms

## 4.2 Position Detection

The installed sensors were numbered, as shown in Fig. 9. We conducted experiments in which two people passed detection areas in the following order:  $1 \rightarrow 2 \rightarrow 3 \rightarrow 6 \rightarrow 9 \rightarrow 8 \rightarrow 7$ ; when the first person enters area 9, the second person goes into area 1. Walking speed is 0.5 m/s. And output waveforms are processed using the position detection technique shown in Section 3.1.

Sensor output waveform is shown in Fig. 10, and processing results are shown in Fig. 11. If a peak position is seen, it has passed in order. However, if the peak appeared at the same time as the sensor separated in position, distinguishing two or more individuals is possible.

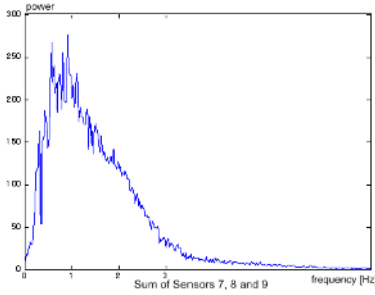
## 4.3 Height Detection

We conducted position detection experiments using sensors 7, 8, and 9. The person passes under the sensor in a straight line. At this time, speed and height were changed and the following data number patterns were extracted. (“Adult” represents state of standing. “Child” represents state of crouching down)

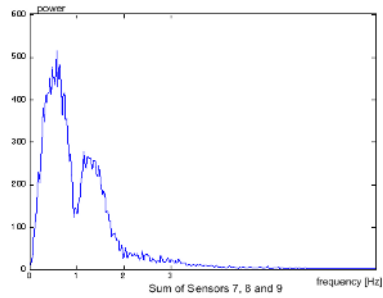
- Adult - Normal Speed (0.8 ~ 1.0 m/s)
- Adult - Slow Speed (0.3 ~ 0.5 m/s)
- Child - Normal Speed (0.8 ~ 1.0 m/s)
- Child - Slow Speed (0.3 ~ 0.5 m/s)

We investigated the accuracy of distinguishing the above pattern by the proposed technique. For example, power spectrums are shown in which the adult and the child are in normal speed. (Figs. 12 and 13) Power spectrums of slow speed are shown in Figs. 14 and 15.

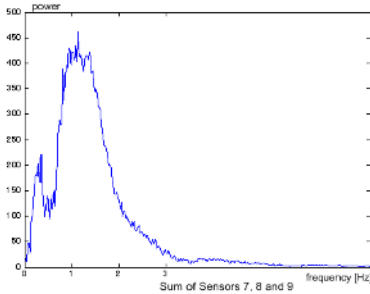
Comparisons of the power spectrums of the four patterns showed the following features. In cases of normal speed, a power of 2 ~ 3 Hz exists to some extent in the adult figures. On the other hand, such figures are very low in the data for children. In cases of slow speed, in the figures of both adults and children, the high power domain is generally low. As mentioned above, the range of *Field I* and *Field II* was set up as follows.



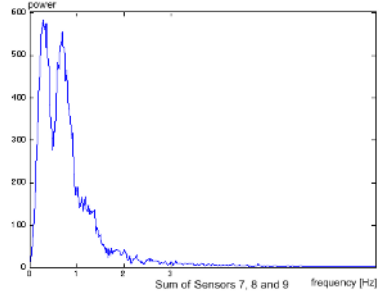
**Fig. 12.** Power Spectrum of Adult - Normal Speed



**Fig. 13.** Power Spectrum of Child - Normal Speed



**Fig. 14.** Power Spectrum of Adult - Normal Speed



**Fig. 15.** Power Spectrum of Child - Normal Speed

- Normal Speed: *Field I*: 0 ~ 1 Hz, *Field II*: 2 ~ 3 Hz
- Slow Speed: *Field I*: 0 ~ 1 Hz, *Field II*: 1 ~ 2 Hz

We performed evaluations that checked the accuracy of the proposed techniques using the examinees of Table 1. Results are shown in Table 2. The probability of making mistakes in height difference judgments is low, even when making mistakes in speed judging. Accuracy is improved by changing the do-

**Table 1.** Height of Examinees

Examinee	Standing	Crouching
ex.1-male	174	114
ex.2-male	180	123
ex.3-male	168	95
ex.4-male	182	113
ex.5-male	172	115
ex.6-female	165	110
ex.7-female	162	110

**Table 2.** Evaluation Results

	Speed Mistakes	Height Mistakes
Adult - Normal	1/8	0/8
Child - Normal	3/8	1/8
Adult - Slow	3/8	2/8
Child - Slow	1/8	0/8
Total	8/32	3/32
Accuracy	75%	91%

main of *Field* I or II after speed judging. Therefore, even if it made a mistake in speed judging, it turned out that the judgment of height difference itself is seldom affected. The accuracy of height difference detection was realized about 90% of the time.

## 5 Conclusions

In this paper, we proposed a technique for other uses of pyroelectric sensors than a switch. Such sensors have the following features: cheap, no worries about privacy or generation of electric waves.

When this sensor is installed in a ceiling in the shape of a lattice, the following phenomena can be detected. 1) Detection in the radius of about 0.8 m is possible in the position of the human being in the room. 2) The position detection of two people is also possible to some extent. 3) For children and adults in whom height greatly differs, detection (adults or children) is possible.

Since this sensor does not have such a quick response speed, it is difficult to use when we need high accuracy. Our future targets include more accurate detection techniques to take advantage of this sensor's features. A concrete service proposal is determined and cooperation with other sensors is sought.

## References

- [1] Shinya Tamano, Ryusuke Nakatani, Shigeo Kaneda, and Hirohide Haga: Position discernment technique using multi pyroelectric sensors. 66th IPSJ, 4H-7, March, 2004, In Japanese.
- [2] Kazutaka Okamoto: Pyroelectric sensors with a self-diagnostic function HORIBA Technical Report, No. 11, Sept., 1995. In Japanese.
- [3] Koichi Matsumoto and Kazutaka Okamoto: Pyroelectric Sensors HORIBA Technical Report, No. 7, July 1993. In Japanese.

# ENME: An ENriched MEdia Application Utilizing Context for Session Mobility; Technical and Human Issues

Egil C. Østhus<sup>1</sup>, Per-Oddvar Osland<sup>2</sup>, and Lill Kristiansen<sup>1</sup>

<sup>1</sup> Dept. of Telematics, NTNU, Norw. Univ. of Science and Technology,  
O.S. Bragstads Pllass 2A, NO-7491 Trondheim, Norway  
{egilconr, lillk}@item.ntnu.no

<sup>2</sup> Telenor R&D, Otto Nielsens vei 12, N-7004 Trondheim  
per-oddvar.osland@telenor.com

**Abstract.** We look into the combination of a SIP application (IP based multimedia telephony) together with a context-aware smart environment. We start by describing a scenario where it is highly relevant to use such a combination. The combined application is called ENME, and is managing and moving communication sessions based on user context. We are realizing the service with SIP REFER and SIP extensions. We also discuss briefly other solutions such as 'virtual terminals', and we identify pros and cons of the different solutions. The application is implemented, and runs on a model railroad system, but the context model itself is more general. A next step would be to deploy the application in other smart environments, and we look briefly into a hospital environment. We end the paper by identifying some human issues for the service to work properly, and relate these issues to the technical solutions.

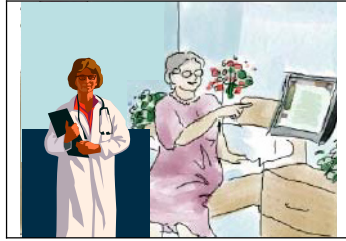
## 1 Introduction

Pervasive computing, a term long spoken of, now seems to slowly become a reality. In visionary descriptions, pervasive computing has been depicted as an environment where computing resources are integrated into more or less every device and physical object. These resources are naturally online, mobile and communicating to fulfill specific tasks.

In our paper we combine pervasiveness and context with an IP multimedia telephony application realized by the SIP protocol from IETF. We focus on a (value added) service ENME that manage communication sessions based on user context.

*A scenario illustrates the use of such an application:* A (human) user moves from a low capability zone into a high capability zone during an ongoing communication session. (The movement may be detected with various location technologies.) The new zone has higher capacity such as a (bigger) screen, video possibilities and so forth. We assume that the user is alerted about the new features and decides to move (parts of) his ongoing session to this new device(s).

The scenario serves as a basis for an implementation of an application realizing one of the solution proposals. The application is given the name ENME. Our implementation is done on a model railroad system in order to simulate mobility between zones, but here we illustrate the use of this application in a hospital setting.



**Fig. 1.** The physician utilizes a bigger screen when entering a patient room

We will look into relevant background work from pervasive computing and context awareness, and combine this with the IP-based multimedia telephony. We will look into different ways to implement the application. The different solutions will be briefly compared. In this paper we identify some issues relating to human factors. We look into those issues having implications on the technical realization in terms of SIP.

We will end the paper by illustrating how ENME may be used in a hospital scenario and some human factors of particular importance.

## 2 Former Work on Pervasiveness and Context

Based on the previous scenario, the following are identified as relevant background material and former work on pervasiveness and context. In the later sections this material will be linked to our implementation and discussion.

### 2.1 Pervasive Computing

Much work on mobile computing, ubiquitous and pervasive computing comes from computer science research.. We have our background in (mobile) telematics / telecommunication, and to us live audio, video and continuous handover of such sessions are natural to look into. In a converged manner we also look into endpoint capable of both 'computing' and 'communication'.

We may note that ICT can be invisible in several ways, e.g. by being integrated into other devices such as refrigerators, eye wear (glasses) etc, or by being mentally integrated as a natural human tool.

Satyanarayanan [1] points out four research thrusts in connection with pervasive computing:

**Effective use of smart spaces.** A space is a meeting room, a corridor or a well-defined area.

**Invisibility.** Weiser's ideal that the computers disappear from the users' consciousness.

**Masking Uneven Conditioning.** The deployment of pervasive computing into the environment is depended on non-technical factors such as organizational structure, business models and economics.

**Localized Scalability.** The number of smart spaces increases, The computational power in those spaces increases as well. The presence of multiple users will further complicate the problem. [1] regards scalability as a problem in pervasive computing. Coming from more of a telecom side we might reformulate it as an issue to be solved. (Telecom has a long history in handling scalability.)

For a more detailed overview of pervasive computing and related topics, see e.g. Satyanarayanan [1].

## 2.2 Context

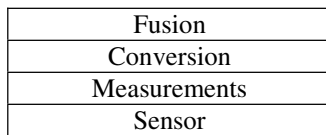
In a pervasive environment, context awareness is an important factor. At the same time, the definition of context is somewhat unclear. There have been several papers published, either trying to define context awareness or presenting new models for context. When this is said, no standards have been proposed. In this paper we use a definition proposed by Dey [2].

*“Context is any information that can be used to characterize the situation of an entity. An entity is a person, or object that is considered relevant to the interaction between a user and an application, including the user and application themselves.”*

This definition points out that both the application and the user is important when it comes to context-aware computing. In addition, it points out that the device executing the application is important as well as the nearby devices.

### 2.2.1 Managing Context Info

Using the above definition, it is obvious that context spans over a wide area of information. Mostéfaoui et al [3] suggest splitting the context into three categories: *Sensed* context, *Derived* context and *Explicitly provided* context.



**Fig. 2.** The context stack [4]

To ease the handling for the context information, Li [4] propose a context stack as shown in Figure 2. The different characteristics of the context information are taken into account in this model. This layering model makes it possible to handle the fact that context information has many alternative representations. The context application using this context stack does not need to know about this, a well-defined interface between the stack and the application hides such details. The context stack is also suitable for derived context, if the sensor layer is thought of as a logical sensor.

### 2.2.2 Context Models

When the context is to be incorporated as a part of a computing environment, the context entities have to be represented in a manner suitable to the execution logic. Different models have been proposed. E.g. Henriksen et al [5] introduce an object-

oriented approach. They suggest dividing the context information into *persons*, *devices* and *channels*. The association between these entities is further subdivided into static and dynamic associations. (See also Figure 3(right part) in section 4.2 for details of their model)

### 3 Enabling Technologies

This section gives a brief overview of some technologies relevant for realizing context-based session management.

One may envision that a high capacity zone may be determined either by some sort of location technology, or some sort of service discovery protocol. We refer to the overview paper by Helal [9] for more information on service discovery.

The positioning used in our implementation is based on Radio Frequency Identification (RFID).

The rest of this section a brief introduction to SIP and SDP. More info on SIP may be found e.g. in the book [6].

Session Initiating Protocol (SIP) is a protocol used to establish and maintain a session. SIP is defined by IETF in RFC 3261[7], often referred to as “Baseline SIP”. The basics components in SIP are the SIP UserAgent (UA), SIP Proxy, SIP Registrar server and SIP redirect server. The user is typically interacting with the SIP UA. The user is identified with a SIP address, that looks just like an email address “sip:userA@item.ntnu.no”. When a user wants to use a SIP UA, he has to register himself with the SIP Registrar. Dialog is a key concept in SIP. One or more dialogs can be a part of a single session.

Session Description Protocol (SDP) [8] provides the receiver with information about the multimedia session and makes the receiver able to participate in the session if desired. A multimedia session consists of a set with media streams that has certain duration. SDP is carried within an SIP message, and typically describes session name and purpose, time(s) the session is active, the media comprising the session, and information to receive those media (addresses, ports, formats and so on).

## 4 The Design of the ENME (Enriched Media) Application

The design of the ENME application is based on the work and principles as presented in sections 2 and 3.

In this section we start by presenting a minimal yet adequate model for applications that manage communication sessions based on user context. Then we present the ENME service and the realized system.

### 4.1 Entities and Relations

Our model consists of a set of basic entities: User, Zone, Device and Session. These entities are described in Table 1 and the relations between them are sketched in Figure 3.

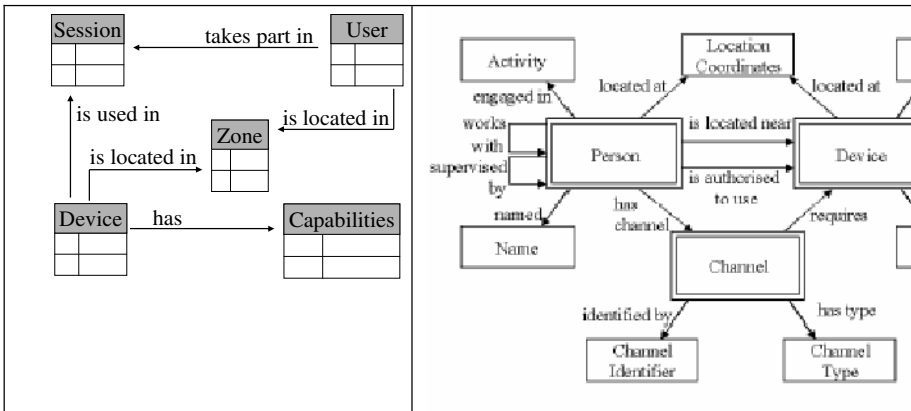
Communication is between users, the technical part of the communication being handled by devices. Users move between various zones, and the access to devices may change when a user moves to a new zone. Devices may be stationary within a

zone, or may move from zone to zone. A user or device may be in one and only one zone (at a time). A zone may have zero or more users and devices. In our implemented system we assume that all users in a zone have access to all devices in that zone, but authorization mechanisms can be supplied to relax this assumption.

**Table 1.** Entities

Entity	Comment
User	A person with access to use the offered application
Zone	A geographical area, e.g. a room or inside a booth. A user and a device are located inside a zone.
Device	A terminal, both public and nonpublic available. A device is described by its capabilities (ability to support video and voice, screen resolution, speakers, related codecs, etc). Sub-entities: High/low capability device. (In the implementation only available codecs are looked at.)
Session	A session is a communication between two or more parties. A session may consist of zero or more dialogs. A dialog will comprise media transfer. A zero-dialog session consists of only signaling, but no media transfer. This is according to SIP [7], and is further described in 0

The data model in Figure 3 (left) excludes the details about available attributes. For sessions, it is important to have information about both the requested (wanted) media description, and those currently used. These values are available in the SDP message (but the wanted description may not be kept by the endpoint (UA), hence we need to keep it in our context model). In addition is it important to include the session identifier. Its capabilities and the zone it currently is within describe the devices. Contact information is also important information. The session information is dynamically updated as sessions are accepted and terminated.



**Fig. 3.** Entities and relations. Left: Our model. Right: The model from [5]. We may roughly compare user - person and session – channel.



## 4.2 ENME: Service (Application) Logic

The main objective behind the model is to create an application that manages communication sessions based on context. More precisely, we want to take into account the devices a user has available, and engage the session at the device that best suits the users' requirements.

As the user move, device availability may change. (We assume availability of 'public' high capability devices available in the user's environment.) The model should also facilitate session mobility, i.e. to move the session to a new device if it is better suited than the one currently in use. To decide if a newly available device is better suited than then ones in use, it is necessary to keep overview of service descriptions requested at the beginning of the session. This leads to a service logic as briefly described in Table 2.

**Table 2.** Service logic

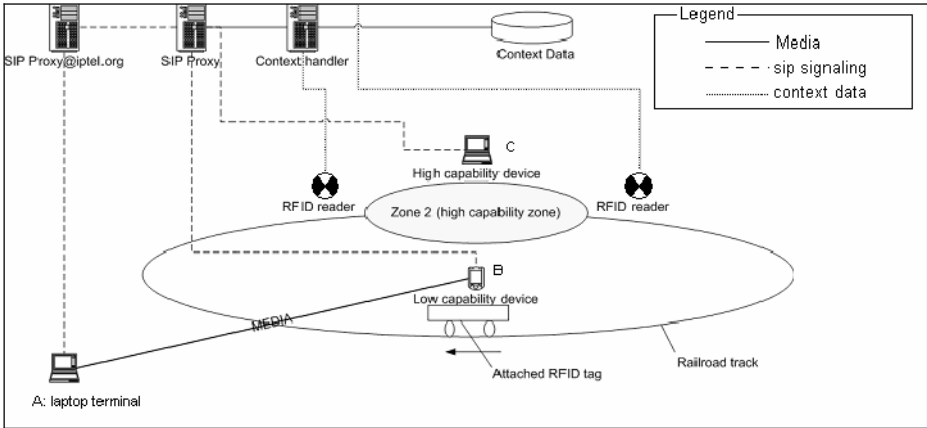
- |   |
|---|
| <ol style="list-style-type: none"><li>0. Assume there is an ongoing communication session between two or more users, the service maintain knowledge about users preferences when it comes to service description, e.g. voice only or video.</li><li>1. If a user moves to a new location, check for available devices. If there are devices that are better than the ones in use, and match user's preferences, proceed to Step 2. Otherwise return to Step 0.</li><li>2. Request the user if he/she wants to move the session to a new device. Proceed to Step 3 if positive answer, return to Step 0 otherwise.</li><li>3. Move (parts of) the session to new device.</li></ol> |
|---|

## 4.3 System Overview and Design

A fundamental principle in our implementation has been to use standard protocols for service parts that involve devices, and develop a network-centric service that keeps as much as possible of the developed logic in a controllable environment. This enables us to use off-the-shelf software for terminal-related parts of the service (with some minor modifications). The service itself is located at a centralized application server. A deployment overview is given in Figure 4 (next page).

The context stack described in section 2.2.1 is partly used, as we separate the RFID (sensor) from other context handling. This will enable RFID to be used in many different applications.

REFER [10] is a SIP extension requesting the recipient to refer to a provided source. Call transfer is one service that is enabled by REFER. What makes the use of REFER somewhat complicated for session movement, is the fact that the REFER RFC does not require the originally dialog to be automatically terminated. REFER requests the receiver to contact an additional source, not to actually move an ongoing session. For the session to be moved, it requires the (human) user to active take part by terminating the originally dialog. For session movement it is convenient that the user terminal performs this task automatically. In order to achieve this, it is necessary to be able to specify different ways to use REFER.



**Fig. 4.** Deployment overview. The prototype is implemented on a model railroad system (as indicated) A, B and C will be referred to later in the text.

ENME has focus on demonstrating SIP for session management. Issues such as context distribution are left out of this paper. In order to be able to implement the service logic as described in Table 2 various solutions are possible, and one was implemented. (Some alternatives are discussed in section 5.)

In order to involve the (human) subscriber of the ENME service, we found it necessary to introduce a SIP extension (we call it CCRequest). This will be sent to the B terminal, and will be followed by the SIP REFER (after a dialog window to the real end user. The CCRequest is a SIP extension that requests the receiver to initiate a request.) The SIP extension was needed because current SIP and SIP extensions does not offer this functionality. Details of the SIP flows are left out due to lack of space.

The main steps from Table 2 will be realized as follows: The Context Handler is using a SIP Interface in order to send and receive SIP messages. When the Context Handler is notified that a user has moved to another zone, and if the service logic described in Table 2 executes successfully, it sends a CCRequest to the B terminal in order to notify it of the newly available high capability device. For the ENME application the receiver initiates a REFER message to its corresponding communication partner. After that the message flow is according REFER RFC [10], and the media flow is now between A and C.

## 5 Discussion

There are several ways to implement the service logic described in step 3 in Table 2. We will discuss pros and cons for the implemented solution, as well as for some alternatives. Details of the alternatives in terms of SIP messages etc. will not be described here due to lack of space, but we will give some evaluation on some of the solutions. The discussion is focused around human factors. Some technical issues are also mentioned.

### 5.1 The Implemented Solution: Some Discussion

The implemented solution puts the (human) user communicating using device B in control over the service execution. Since B is the subscriber of the ENME service this makes perfectly sense. The REFER RFC will establish an Event subscription forcing the A terminal to notify B the outcome of the session movement. This way the B user is aware of why the session handover fails if it does. Such information is considered crucial for being able to create a user-friendly service. These are 2 important pros with this solution. On the more technical side: The negative side is that we needed a SIP extension. The positive side is that A only needs to support the basic SIP/SDP messages and the well established REFER method for ENME to work.

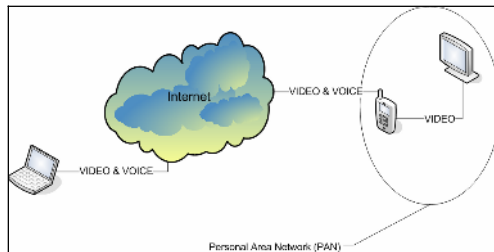
### 5.2 Alternative Solution 1: Some Discussion

This alternative is to have the ContextHandler/SIPinterface sending a REFER message to the A terminal. This raises some problems, and we will describe two of them here. The (human) user communicating using the B terminal is the subscriber of the service, not the user at terminal A (who may in fact be unaware of the existence of ENME application in the general public case). Consequently should the (human) B user be queried whether the session should be moved or not. On the more technical side: Also A (who is not subscribing to this service) will need updated software in this case.

### 5.3 Alternative Solution 2 ('Virtual Terminal'): Some Discussion

'Virtual terminal' is a concept to combine different devices in order to deliver a service that is capable of deliver richer media than any of the devices would manage alone.

One approach is send both video and voice to the low capability device, and have the low capability device forward the video part of the stream to the high capability device using a Personal Area Network (PAN). In this case the low capacity device needs not display the video, but it needs to forward it. This solution puts further requirements to the user terminal, both in term of service discovery and video reception/forwarding.



Interactive conversational multimedia showing the colleague's face puts strong requirements both on synchronization and time delay. By creating virtual terminals, problems with synchronization and timing might become an issue. We may notice that this issue does not occur for data applications, but is crucial for many uses of video streams, since it is a fact that lip-out-of-synchronization is more disturbing than no video at all.

Talking to a colleague and at the same time looking at an independent video stream will however have fewer requirements on the synchronization.

## 6 ENME in a Hospital Setting: Some Issues

We have already indicated (Figure 1) that ENME may be deployed in a hospital setting, via high capacity screens on patient rooms, in corridors, in offices or in special ‘multimedia booths’. In these cases video telephony may be relevant, but also cases of voice telephony enhanced with data applications are relevant (like ‘joint looking’ at a patient journal). It might also be relevant to look at video of say ECG, while having voice conversation to another person. We will now identify some human issues relevant for the ENME service.

But first we will comment on the use of a location infrastructure inside a hospital. Our system uses RFID sensors. As pointed out already in 4.3 our solution allows for many applications to use the same infrastructure. Other applications that may take advantage of location might be drug management, as proposed in [11].

### 6.1 User Involvement or User Disturbance?

The ENME application as described provides help to its users to establish as rich media session as possible at any time. The naïve assumption is that all users always want the richest media available, *but this is not necessarily the case*. Følstad et al. [12] has studied what criteria were used to decide what communication service to use for different tasks. This field study showed that *the users where conscious what communication service was most suitable for each task*. Hence it is obvious that adding video to an ongoing communication service may not be wanted in all cases, and our solution supports this. It may also be useful to separate voice-with-possible-enhancement-telephony from voice-and-always-video-telephony. ENME may be used with both, but the disturbance of changing terminal must be considered for the latter, and an ENME on/off button may easily be added.

### 6.2 One Terminal or Many Terminals?

The Knowmobile research project gave PDAs, laptops and GSM-phones to physician students carrying out work in hospitals. In [1] they write: “The multidevice paradigm leads to problems connected to the use, design, harmonization, [...] of various devices”. This shows that we are facing challenges in the design of an application like ENME.

Other question like: Shall the GUI for ENME application control move to the bigger screen, or shall it stay on the handheld device at all times? This needs to be studied further, and may require some technical studies (at SIP level) as well.

## 7 Conclusions and Further Work

By adding context awareness to a communication service, it is possible to create an application that informs the (human) user if a more appropriate device is available nearby. The session can in this situation be moved to the best-fitted device upon

acceptance from the user. SIP and SIP REFER was used to realize this service. We also pointed out that the subscriber of the service has to be involved in the service execution.

We also analyzed some human factors, and found that humans may not always want a pervasive environment, as pervasive may in fact be considered ‘invasive’.

In chapter 6 we identify some human issues relating to disturbance and confusion. These issues are important in all cases, but become more important in cases where the communication itself is not the main task, just a tool supporting the real tasks and activities carried out by the users. This is typically the case in a hospital. Thus it seems relevant to put more focus on the *tasks* that the user is carrying out, not just the (technical) channels and the devices. This fits in with the findings of [12]. Also the relations between the colleagues may be of importance. This leads us to further organizational issues. This is however left out of this paper.

The application is not analyzed when it comes to privacy issues and security. Security is a particular issue in a hospital setting, but we believe it is important also for ENME as a service for personal use in public settings. Thus this needs further work. Organizational and cultural issues must be studied as well, in particular in the hospital setting.

## References

- [1] Satyanarayanan, M., “Pervasive Computing: Vision and Challenges”, IEEE Personal Communications, August 2001, pp 10 – 17
- [2] Dey, A. K., “Providing Architectural Support for Building Context-Aware Applications” PhD thesis, College of Computing, Georgia Institute of Technology, 2000
- [3] Mostéfaoui, G. K., Pasquier-Rocha, J., Brézillon, P., “Context-Aware Computing: A guide for the Pervasive Computing Community”, proceedings of the ICPS’04
- [4] Li, Wei, “A Service Oriented SIP infrastructure for Adaptive and Context-Aware Wireless Services”, proceedings of the 2nd International Conference on Mobile and Ubiquitous Multimedia, 2003 (Norrköping, Sweden)
- [5] Henriksen, K., Indulska, J., Rakotonirainy, A., ”Modeling context information in pervasive computing systems”, proceedings of Pervasive 2002 (Zürich, Switzerland): 167-180, 2002
- [6] Johnston, Alan B. SIP: Understanding the Session Initiation Protocol, Artech House, 2001
- [7] Rosenberg, J., Schulzerinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., Schooler, E., “SIP: Session Initiation Protocol” IETF Networking Group, RFC3261, <http://www.ietf.org/rfc/rfc3261.txt>
- [8] Handley, M., Jacobson V., “SDP: Session Description Protocol”, IETF Networking group, RFC 2327, <http://www.ietf.org/rfc/rfc2327.txt>
- [9] Helal, S., “Standards for Service Discovery and Delivery”, IEEE Pervasive Computing Vol 1(3), July – September 2002, pp 95 – 100
- [10] Sparks, R., “The Session Initiation Protocol (SIP) REFER Method”, IETF Networking group RFC 3515, <http://www.ietf.org/rfc/rfc3515.txt>
- [11] Lindeberg, L. and L.Kristiansen, “How a context-aware resource planner for drugs can increase quality in health care” Presented at SHI2005, Aalborg, Denmark, August 2005
- [12] Følstad, A. et al., ”Fitness-for-Purpose of Person-Person Communication Technologies”, IST11577/SEF/DIS/DS/Pub/004/b1, October 31st, 2002
- [13] Gallis, H., Kasbo, J. P. and Herstad, J., “The Multidevice Paradigm in Knowmobile - Does one size fit all?” Proc. of IRIS24, ed. by Bjørnestad, S. , Moe, R.E., Mørch, A.I. and Opdahl, A.L. , (2001) pages 491-504

# DartDataFlow: Semantic-Based Sensor Grid

Huajun Chen, Zhiyong Ye, and Zhaohui Wu

College of Computer Science, Zhejiang University,  
Hangzhou, China  
{huajunsir, yezhy, wzh}@zju.edu.cn

**Abstract.** We propose DartDataFlow, a semantic-base Sensor Grid to manage sense data. In our system, the data and subscriptions are all represents as RDF graph, so we provide a RDF Graph Filter to filter RDF graph to meet subscriptions. And also, we design an intelligent data dissemination framework to support different cases of data dissemination. This system integrates Semantic Web technology into Sensor Grid and can be used in intelligent transportation systems, military, weather forecast, environment monitor, etc.

## 1 Introduction

Today, plenty of sensor nodes in sensor grid generate rapid, continuous and large volumes of stream data .It is one of the most important issues how users communicate and understand data when sharing and receiving data from sensor grid [1]. A solution for this issue is for each sensor to publish data schema based upon a shared ontology using Semantic Web technology [2]. Therefore, when a sensor node publishes data, users that receive data will be able to interpret the contents without ambiguity. A solution for this issue is for each sensor to publish data schema based upon a shared ontology using Semantic Web technology. Therefore, when a sensor node publishes data, users that receive data will be able to interpret the contents without ambiguity.

On the other hand, publish/subscribe (pub/sub) [1, 3] is a loosely coupled communication paradigm for distributed computing environments. In pub/sub systems, publishers publish data to brokers in the form of event, subscribers subscribe to a particular category of data within the system, and brokers ensure the timely and reliable delivery of published data to all interested subscribers. The advantage of pub/sub paradigm is that information producers and consumers are full decoupled in time, space and flow, so it is well suitable for large-scale and highly dynamic distributed systems.

In this paper, we introduce a semantic-base Sensor Grid named DartDataflow (DDF) to fuse and disseminate sensor data. In DDF system, publishers and subscribers share a RDF-based ontology. This releases the constraints on value-based publish/subscribe systems that publishers and subscribers must share the same data schemas. Different from relational pub/sub systems, the data correlation in our system is transparent to subscribers.

The remainder of the paper is organized as follows. In Section 2, we discuss related work. In Section 3, we introduce Resource Description Framework which is the preliminary of this paper. In Section 4, we introduce the system architecture of the DDF system. In Section 5, 6, we describe Fusion Layer and Dissemination Layer detailedly in the DDF system. Finally, in Section 7, we conclude the paper with a summary.

## 2 Related Work

In the past several years, many data dissemination schemes have been proposed for sensor networks. Based on where data generated are stored, these schemas [4, 5] are divided into three categories: local-storage, data-centric storage-based and external-storage.

- 1) External storage-based data dissemination relies on a centralized base station, which is external to the sensor network, for collecting and storing sensing data. In this schema, data must be sent back and forth between the sensors and the base station.
- 2) Data-centric storage-based data dissemination stored the sensing data at certain nodes within the network. In this scheme, data are still pushed in a predefined manner regardless of queries.
- 3) Local storage-based data dissemination is the dissemination schemes which a source sends data to a sink only when the sink has sent a query for the data. These schemes need a sink-source matching mechanism to facilitate a sink to find the source holding the data of interest.

These schemes include directed diffusion [4], two-tier data dissemination [5], etc.

Pub/sub [3] systems are generally divided into two categories: subject-based and content-based.

- 1) The earliest pub/sub systems are subject-based. In those systems, each data belongs to one of a fixed set of subjects (also called topics, channels, or groups). Publishers are required to label each data with a subject name; consumers subscribe to all data under a particular subject. The techniques for subject-based pub/sub systems have already matured and there are many successful products and specifications such as TIB/Rendezvous [6] from TIBCO [6] and MQSeries from IBM [7].
- 2) The topic-based publish/subscribe variant represents a static scheme which offers only restricted expressiveness. In consequence, as improvements to topic-based solution, content-based publish/subscribe systems are proposed. In these systems, data are no longer divided into different subjects. The subscriber defines a subscription condition according to the internal structure of data; all data that meet the condition will be sent to the subscriber. The internal structures of data are defined as data schemas. Compared with the subject-based pub/sub systems, the content-based systems are more expressive and flexible; it can enable subscribers to express their interests in a finer granularity. Known prototype systems include Gryphon [8], Siena [9], etc.

However, if users are going to share and receive data from heterogeneous and highly dynamic sensors in the sensor grid, then it must be able to communicate and to understand this data. Obviously, these schemes as introducing above are not sufficient for this application. Our goal is to introduce the Semantic Web technologies [5] into the pub/sub system to support data fusion and data dissemination in sensor grid.

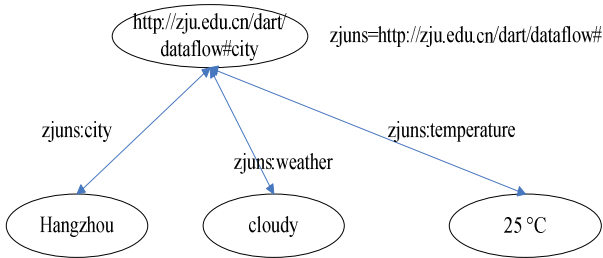
## 3 Resource Description Framework

The Resource Description Framework (RDF) [5] is the basic technology of our system. We first introduce it simply. RDF is a language for representing information

about resources in the World Wide Web .The purpose of RDF is to define the specifications for supporting the use of Metadata about Web resources. This should be accomplished in a manner that will allow a uniform way for the machines to understand and process the information given, together with the flexibility of describing the information for all information publishers. The main concepts of the RDF specifications are the description using Metadata of any kind of resources that can be named via a URI (Uniform Resource Identifier), the interoperability between applications that exchange machine-understandable information on the Web. The specifications also address the ability to enable automated processing of Web resources and finally the capability for different application communities to define their own Metadata Property Set that best serves their needs.

Any expression in RDF is a collection of triples, each consisting of a subject, a property and an object. A set of such triples is called an RDF graph. The nodes of an RDF graph are its subjects and objects. The assertion of an RDF triple says that some relationship, indicated by the property, holds between the things denoted by subject and object of the triple. The assertion of an RDF graph amounts to asserting all the triples in it, so the meaning of an RDF graph is the conjunction of the statements corresponding to all the triples it contains.

Example 1 is a small chunk of RDF in RDF/XML corresponding to the graph in Figure 1:



**Fig. 1.** An RDF graph describing of the weather of Hangzhou

```

<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:zjuns="http://zju.edu.cn/dart/dataflow#">
  <zjuns:City rdf:about="http://zju.edu.cn/dart/dataflow#city">
    <zjuns:name>Hangzhou</zjuns:name>
    <zjuns:weather>Cloudy</zjuns:weather>
    <zjuns:temperature>25 °C</zjuns:temperature>
  </zjuns:City >
</rdf:RDF>
  
```

**Example 1.** RDF/XML describing of the weather of Hangzhou



## 4 Architecture

In sensor grid, data published by heterogeneous sensors differs greatly in formats and semantics. If the pub/sub system is used as a general infrastructure for sensor grid and support different applications, it should have strong expressiveness, i.e.:

- 1) It should support data in different formats and semantics;
- 2) It should provide a powerful subscription language, so that data consumers can easily express their interest in certain data semantics.

Here, we introduce the Semantic Web technologies into the pub/sub paradigm and propose an Semantic-based Publish/Subscribe middleware named Dart-Dataflow (DDF) for manage sensor data .It’s architecture is as Figure 2 shown.

Physical layer is composed of sensor nodes deployed in sensor grid. These sensor nodes collect dynamic and real-time traffic data, weather data, geography data, road surface data and so on. This layer also provides an interface to disseminate these original data into upper layer.

Fusion layer mainly supports to fuse data from heterogeneous sensors and filter data to meet the subscriber. It includes Ontology Service, Semantic Parser Service, Semantic Subscription Service and Semantic Data Filter Service. Services in this layer are provided for sinking data from sensor nodes, querying data schema based on the global ontology and parsing it into RDF-based data and then filtering data to meet the subscribers interesting.

Dissemination layer provides an intelligent and flexible mechanism to disseminating data to the subscriber. It includes Negotiation Service, Performance Monitor Service and Dissemination Service. In this layer, data dissemination is multiple protocols implementation, migration support, easy extensibility.

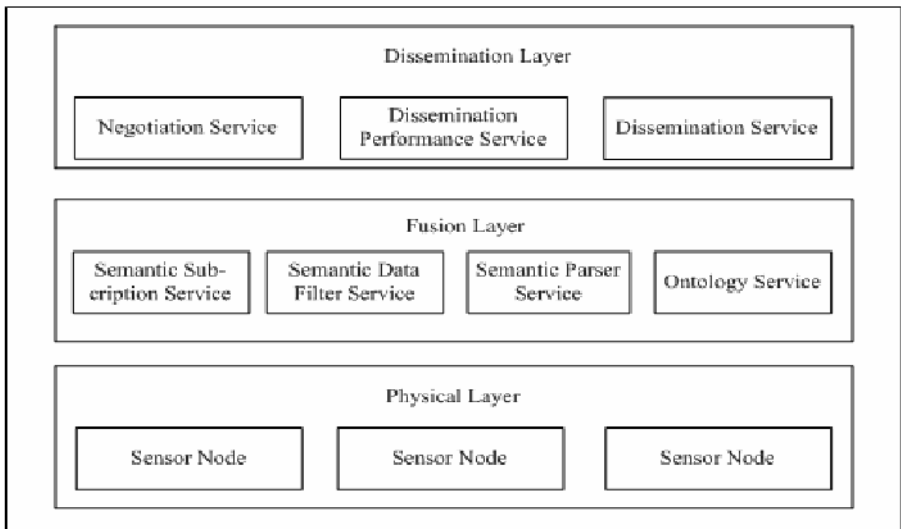


Fig. 2. Architecture of DDF

## 5 Fusion Layer

In DDF, data published by sensors and subscription scripts are both represented as RDF graphs, which is a kind of directed labeled graph and the system filter data with subscriptions both semantically and syntactically. When data is published, it is firstly converted into a RDF graph using Semantic Parser Service based on ontology before further processing. And the same, subscriptions are specified by subscriber and are also represented as RDF graph patterns in Semantic Subscription Service. Then Data Filter Service filters the data to meet the subscriber. As follows we introduce two keys in Fusion Layer.

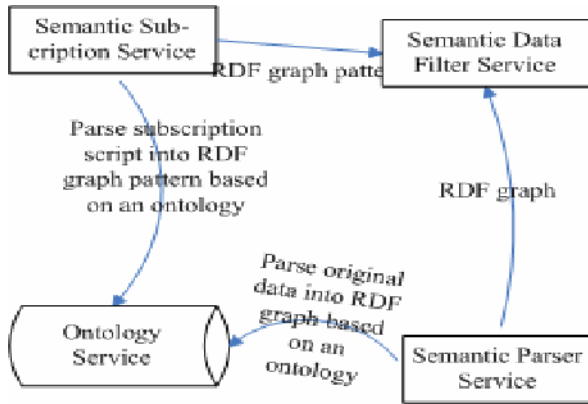


Fig. 3. Mechanism of Fusion Layer

### 5.1 Semantic Subscription Language

In DDF system, data are all represented as RDF graphs; the subscription is in fact a graph pattern which specifies the shape of the graph as well as the constraints on some nodes and arcs in the graph. So we must define a subscription language to support semantic subscription. Here we define a subscription language based on a number of query languages such as SPARQL [10], RDQL [11].

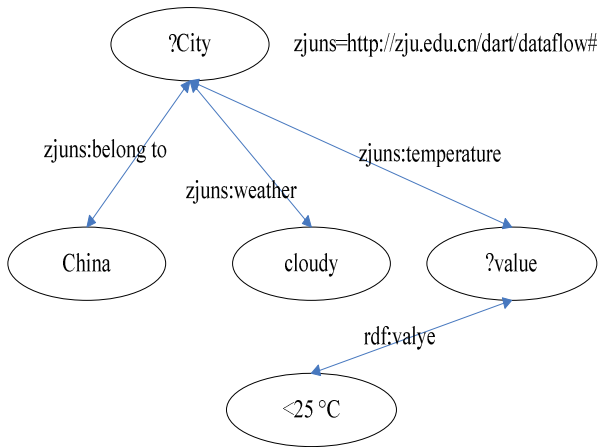
Our subscription language is provided for accessing RDF graphs. A subscription mainly consists of two parts, the Select clause and the Where clause. The Select clause names the variable of interest to the subscriber; and the Where clause has one or more triple patterns which are composed of a subject, predicate and object delimited by parentheses, and for confining the value of object, the Where clause always defines some filter expressions.

For example, in the Weather Forecast system, if someone want to know all cities which are in China with the temperature is below 25 centigrade thermometer and the weather is cloudy, he can express it as:

```

Select ?city
Where (?city zjuns:belongto China)
      (?weather zjuns:weather cloudy)
      (?temperature zjuns:temperature ?value)
And ?value<25
Using zjuns for <http://zju.edu.cn/dart/dataflow>
    
```

**Example 2.** An example of a subscription language



**Fig. 4.** An example of graph for the subscription language

As above mentioned, each subscription can represent as a RDF graph pattern in our system, so preceding subscription can be represent as the corresponding RDF graph pattern in Figure 4.

### 5.2 RDF Graph Filter

In the DDF system, we must filter the incoming data to meet the subscription .We design a RDF Graph Filter to support this task. Subscriptions and data are all represented as graphs in our system, and if every node and arc in a data graph can be mapped to a corresponding node and arc in the subscription graph, the data is said to meet the subscription, otherwise the data can not meet the subscription. Therefore, the filtering problem is a specific kind of graph isomorphism problem.

Following we focus on our graph isomorphism arithmetic. For example, our subscription graph  $G(s)$  is as Figure 5 shown.

We suppose a RDF graph can show as:

$G = (H, V, E)$ , where

$H = \{H\}$ ,  $H$  is the home vertex;

$V$  is the set of the vertexes of the RDF graph and while a vertex has no outgoing edge, we define it is a specific values or an expression;

$E = \{R_i = E(u,v): u,v \text{ belong to } V, \text{ and the direction of the edge is from } u \text{ to } v. \}$ .

Then we can represent  $G(s)$  as

$G(s) = (R_s, V_s, E_s)$ , where

$R_s = \{H\}$ ,

$V_s = \{H, A, B, C, D, E\}$ ,

$E_s = \{R_1 = E(H,A), R_2 = E(H,B), R_3 = E(H,C), R_4 = E(B,D), R_5 = E(B,E), R_6 = E(C,E), R_7 = E(D,E)\}$ .

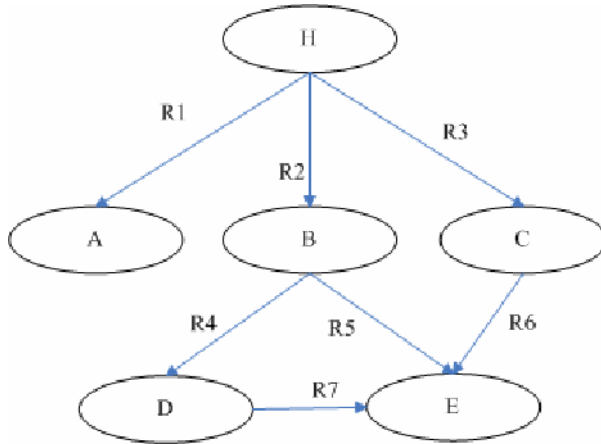


Fig. 5. A subscription graph

**Arithmetic(G):**

- for each vertex  $u \in V_d$
- do flag(u) ← false
- Visit(H)
- for each vertex  $u \in V_d$
- do if flag(u) = false
- return false
- return true

**Visit(u):**

- do flag(u) ← true
- for each  $v \in \text{adj}(u)$
- do if outdegree(v) = 0
- then do if exp(v) = false
- break;
- else do flag(v) ← true
- else Visit(v)
- 

Fig. 6. The filtering arithmetic

When a data which it is represented as RDF graph is coming, we suppose it express as  $G_d = \{H_d, V_d, E_d\}$ . Then we can introduce our graph isomorphism arithmetic as Figure 6 show.

$exp(v)$  is a boolean expression which judge whether the  $v$  in  $G(d)$  meet the constraints of the corresponding vertex in the  $G(s)$ .

## 6 Dissemination Layer

In DDF system, Dissemination Layer is used to disseminate the RDF-base filter data to the corresponding subscriber effectively. Following we introduce the design of the Dissemination Layer detailedly.

As we all know, different transport protocols are suited for different cases. For example, TCP works best where reliable delivery is at a premium but if subscriber can sustain losses in delivery and more concentrate on the latencies, UDP is a better choice. And also, data dissemination in network is influenced by different condition and different network. So in Dissemination Layer, we design more than one Dissemination Services and each service implements a transport protocol.

DDF system provides a Negotiation Service to negotiate the best transport protocol for data dissemination. If negotiate is successful, the system specify the corresponding Dissemination Service to the subscriber.

Considering that the network capability is unstable. So we design Dissemination Performance Service to monitor the dissemination performance. This service can specify a constraint on the performance factors and specify the migration to anther Dissemination Service which implement a more suited transport protocol when this constraint is satisfied. For example where dissemination using UDP is not feasible due to high loss rates user can switch to TCP and similarly, user may switch to UDP by reason of bandwidth and latency constraints.

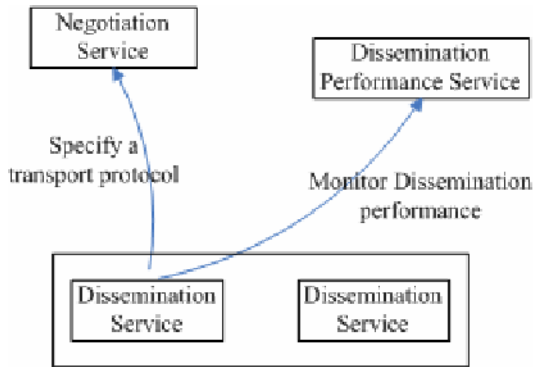


Fig. 7. Mechanism of Dissemination Layer

## 7 Conclusion

DDF system is mainly composed of Physical Layer, Fusion Layer and Dissemination Layer. In Fusion Layer, We define a semantic subscription language to support com-

plex subscriptions. And also, we design a Semantic Data Filter in Fusion Layer to meet the subscriber individual subscription. In Dissemination Layer, we propose a new intelligent data dissemination framework that is appropriate for this semantic-base pub/sub system.

Our future work includes optimization of data filtering arithmetic based on RDF graph to support numerous subscriptions more effectively.

## References

- [1] Bass.T: The federation of critical infrastructure information via publish-subscribe enabled multisensor data fusion. Information Fusion, 2002. Proceedings of the Fifth International Conference on Volume 2, 8-11 July 2002 Page(s):1076 - 1083
- [2] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. Scientific American, 284(5): 34-43. May 2001
- [3] P. Eugster, "Publish/Subscribe," ACM Computing Surveys, vol. 35, no. 2, Jun. 2003, pp. 114-131.2.
- [4] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directeddiffusion: A scalable and robust communication paradigm for sensor networks. In Proc. of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networks (MobiCom 2000), Boston, Massachusetts, August 2000.F.
- [5] Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang. A two-tier data dissemination model for large-scale wireless sensor networks. In Proc. of the Eighth ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom2002), Atlanta, Georgia, USA, September 2002.
- [6] TIBCO Corporation, "TIB/Rendezvous", white paper. <http://www.rv.tibco.com/rvwhitepaper.html>.
- [7] IBM. Internet Application Development with MQSeries and Java, February 1997.
- [8] M. K. Aguilera, R. E. Strom, D. C. Sturman, M. Astley, and T. D. Chandra: Matching events in a content-based subscription system. In: Proceedings of the Eighteenth ACM Symposium on Principles of Distributed Computing (1999) 53-61
- [9] Carzaniga, D. S. Rosenblum, and A. L. Wolf: Design and evaluation of a wide-area event notification service. ACM Trans. on Computer Systems 19(3) (2001) 332-383
- [10] SPARQL: Query Language for RDF: <http://www.w3.org/TR/2004/WD-rdf-sparql-query-20041012/>
- [11] RDQL: RDF Data Query Language. <http://www.hpl.hp.com/semweb/rdql.htm>

# Sentient Artefacts: Acquiring User's Context Through Daily Objects

Kaori Fujinami and Tatsuo Nakajima

Department of Computer Science, Waseda University, Tokyo Japan  
{fujinami, tatsuo}@dcl.info.waseda.ac.jp

**Abstract.** In this paper, we describe an augmentation of everyday artefact called *sentient artefact*. A sentient artefact is expected to capture the user's specific context implicitly and naturally from its original usage since such an everyday artefact has inherent roles and functionalities. Therefore, a context-aware space is built incrementally using the specific contextual information. We show three types of everyday artefact augmentation, and propose a sensor selection framework that allows an artefact developer to systematically identify desirable sensors. Also, we discuss expectations and issues on the augmentation through prototyping.

## 1 Introduction

Ubiquitous computing is envisioning to provide an intelligent environment, however it has not been realized yet. We consider one of the reason is the installation cost. Enabling technologies so far, e.g. location sensing systems and vision-based recognition systems, require complex infrastructures embedded into our environment, which increase the deployment cost. Also, a new type of devices that requires a user to learn its usage might provide him/her with a cognitive burden.

To address these issues, we are working on augmenting daily objects with computing capabilities like sensors and actuators, which capture a user's situation, *context*, and provide him/her its awareness in a natural and implicit way. We call such a daily object a "*sentient artefact*". We use a sentient artefact as a daily object that has inherent functionalities as usual. At the same time, it detects its state-of-use and utilizes the information as an input to a system. For example, a sentient door and sentient chair are utilized as an ordinary door and chair respectively. However, a system perceives a user's *presence* and *state*, e.g. inside the room, sitting on the chair, etc., and changes its behavior according to the contextual information. We believe the sentient artefact approach allows a developer to build context-aware applications easily. Also, from a user's aspect, he/she can utilize a context-aware service implicitly and naturally through the interaction with various sentient artefacts. A sentient artefact is expected to play a key role in realizing a ubiquitous computing environment in a practical way.

We have been prototyping various types of sentient artefacts so far. In this paper, we describe experiences from three of them: a chair, mirror, and toothbrush. We have found that an ad-hoc selection of sensors that depends on a developer's intuition fails to extract proper contextual information. Therefore,

we propose a conceptual framework that allows a sentient artefact developer to select appropriate sensors. The framework utilizes the usage of the artefact and the observable phenomenon through the interaction with a user.

The structure of the paper is as follows. Section 2 shows early prototyping of the three sentient artefacts. In Section 3, a sensor selection framework is proposed through the prototyping. In the discussion section (Section 4), we describe 1) a variety of contextual information beyond the original state-of-use, 2) design issues and our approach towards artefacts integration, 3) structured artefact's information for robust and portable application, and 4) assessment of our approach that employs low-level sensors and combination of sentient artefacts for person identification. In Section 5, we examine existing work regarding augmenting daily objects and extracting context through complex sensing infrastructure. Finally, we conclude the paper with future directions in Section 6.

## 2 Prototyping of Sentient Artefacts

We have been augmenting various everyday objects continuously. Here, we introduce the following three typical artefacts.

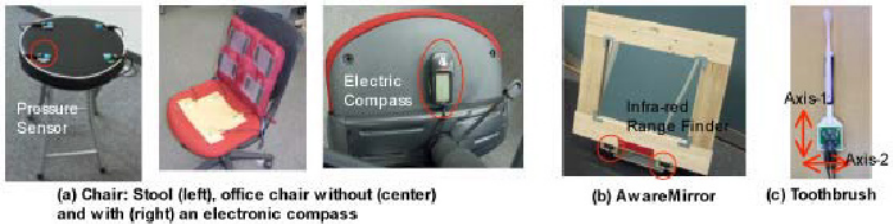


Fig. 1. Prototyped Artefacts

### 2.1 Chair

We use a chair for doing something with sitting on it. The activity at the chair is narrowed in conjunction with the type and/or location of the chair. For example, a chair in an office is utilized for supporting a user's work, while another chair in the kitchen is for eating or chatting. The change of states, i.e. from not sitting to sitting and vice versa, is a trigger for a system to invoke a specific service. We have developed the following three types of *sentient chairs* that can extract different types of contextual information:

1. Only an object's state, i.e. being put by something or not (Fig. 1-(a) left)
2. A person's sitting or not sitting with adverbial states (Fig. 1-(a) center)
3. An approximate direction of the face of a user (Fig. 1-(a) right)

The first case was built by simply augmenting a stool with four pressure sensors on the seat, where it does not distinguish a person from other objects.



In the second case, an office chair with a back seat was adopted, and pressure sensors were attached properly so that the state of legs is extracted. Also, a photo sensor, a pressure sensor and a touch sensor were attached to detect the usage of the back seat. Finally, an office chair was augmented with an electric compass that detects terrestrial magnetism and specifies the direction of the back seat of the chair. These types of chairs suggest the need for an ontology of an artefact to provide applications with robustness and reliability, which will be discussed in more detail in Section 4.3.

## 2.2 Mirror

AwareMirror (Fig. 1-(b)) is an augmented mirror that displays information relevant to a person in front of it on the periphery of his/her sights[4]. The augmentation is natural because we usually “use” a mirror to know our internal state, e.g. health, as well as external state, e.g. looks, through the reflection on the surface of it, which is easily extended to virtual features, e.g. the schedule of the day, the weather forecasting to the destination. AwareMirror suggests an ideal interaction between a person and a computer embedded into our daily lives.

The AwareMirror project have also suggested socially acceptable sensing technology for the artefact augmentation. The state-of-use of a mirror should be the situation that the user is looking at the figure in the mirror. However, we have considered the state as the fact of detecting something in front of a mirror, which is realized by two infra-red range finders to remove a feeling of privacy violation. It is combined with the detection of utilization of a co-located toothbrush to extract more meaningful and accurate information.

## 2.3 Toothbrush

We have also got a survey result that people do not want to share a toothbrush with others, which means that the usage of a toothbrush strongly suggests that its owner is brushing his/her teeth. We have augmented a toothbrush with a two-axis accelerometer to detect the start, end, and the approximate number of brushing (Fig. 1-(c)) [5]. Also in [5], we have utilized a sentient toothbrush as an exclusive activity information source against the user's “sleeping”, which means a sentient artefact is a building block for various applications, rather than a specific application.

## 3 Sensor Selection Framework

The sensors utilized in the prototyping were selected after some trials on a variety of sensors, which depends on a user's intuition. Such an ad-hoc way prevents an application developer from selecting appropriate sensors rapidly and consistently. In this section, we describe a framework to select suitable sensors for respective sentient artefacts in a systematic way. In our framework, the relationship between a user or his/her belonging and an artefact to augment is a key to find an appropriate phenomenon for sensing. The discussions in existing work [3, 6] show the

catalogues of sensors that is utilized to specify a sensor from the phenomena to be measured. However, it lacks of earlier stages in terms of selecting appropriate sensors for artefact augmentation, i.e. what kind of state can be extracted, what kind of interaction is remarkable for the context extraction, what kind of phenomenon can be observed from the interaction, etc. Namely, our framework provides a sentient artefact developer with a systematic way to finally answer a question like “What kind of phenomenon is remarkable to extract the state-of-use of the target artefact most accurately?”. The framework consists of five steps:

**STEP1:** Specify the state-of-use that a developer wishes to extract

**STEP2:** Analyze the usage of a target artefact

**STEP3:** Clarify the observable phenomena in use

**STEP4:** List the candidates of sensors

**STEP5:** Select the most preferable one from various aspects

In the following sections, we describe each step in more detail based on the prototyping results.

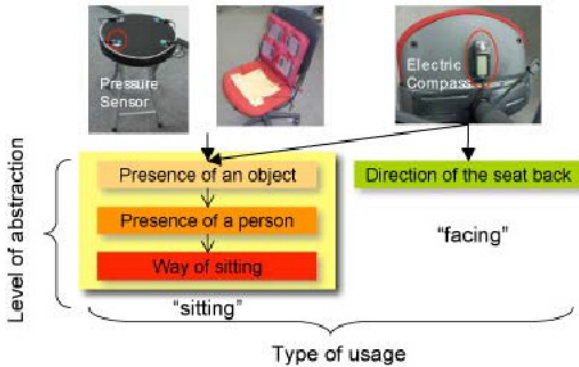
### 3.1 Specify the State-of-Use to Extract (STEP1)

This step is to answer the basic and essential question like “What state-of-use information is required?”. This question can be separated into two detail questions: the level of abstraction and type of information. As depicted in Fig. 2, an office chair can offer two types of usage: the usage as a chair in nature (left side) and that is specific to a chair with back seat (right side). Also, each type might have more detailed usage, e.g. simply putting something on (the presence of an object), let a person sit on (the presence of a person), and let a person relaxed with his/her back lean against the back seat (the way of sitting). Therefore, first of all, a developer has to clarify his/her requirements for the extraction. However, they should be assessed whether the target artefact provides required state-of-use because they might not be provided by the artefacts functionalities.

### 3.2 Analyze the Usage of Artefact (STEP2)

The next step is to analyze the specified usage to answer the question “How to use it? or How to interact with it?”. The result of the analysis classifies the usage into primitives, which include putting/removing, touching/leaving, pushing/pulling, rotating, shaking, approaching/leaving, stroing/extracting, etc. For example, in the case of sitting on a stool depicted in Fig. 1-(a)-left, a user’s hip is “put” on the seat with some force, while a person sitting on an office chair (Fig. 1-(a)-center/right) can “rotate” it and lean on the back seat, i.e. “touching”.

Moreover, it is important to identify the ease of changing of the relationship, that may affect the weight of responsibility of the sensor. In the case of the above office chair, the relationship between a person and the back seat can be changed frequently since he/she might bend and lean, while the relationship between a person and the seat do hardly change. Therefore, to extract the presence of a person, one or more sensors should be attached primarily on the seat, and those of the back seat should be provided to supplement the primary ones.



**Fig. 2.** Heterogeneity of the level of abstraction and type in daily object

### 3.3 Clarify Observable Phenomena (STEP3)

The third step is to clarify the observable phenomena against the primitives. For example, when something is “put” on the surface, there might be physical phenomena like the change of pressure on the surface, the vibration of the surface, making noise, the change of temperature on the surface (in case of a creature), the change of electric capacitance, etc. Although almost the same phenomena can be observed in the case of “touching” on the back seat of a chair, it is difficult to detect the change of temperature because the “touching” happens so often while the speed of changing temperature is slow. This means that “leaving” from the back seat might happen before a temperature reaches at a dedicated level. By the end of this step, the phenomena that contributes to extract the specified state-of-use become clear.

### 3.4 List the Candidates of Sensors (STEP4)

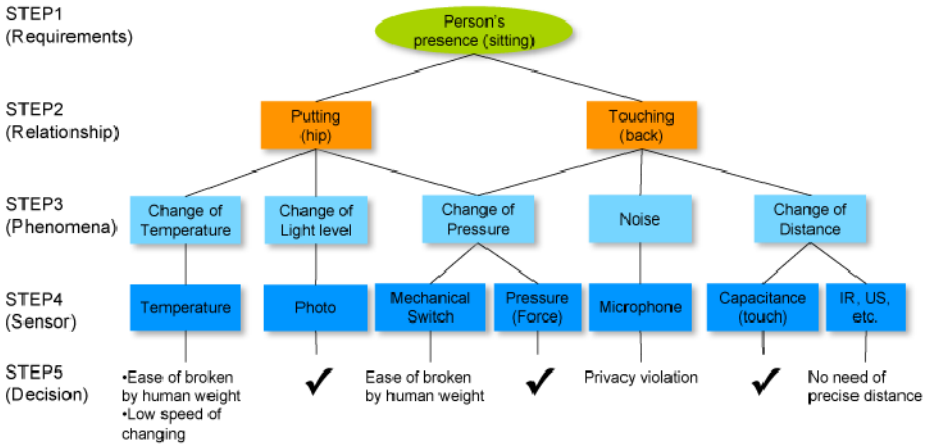
Hereafter, there is the literature to know how to sense a specific phenomenon, e.g. [3, 6]. Namely, the answer to the question like “What sensor can measure a change of force?” is easily found. There may be more than one sensing technology for each phenomenon, which will be identified in the final step.

### 3.5 Select the most Preferable Sets of Sensor (STEP5)

This is the final stage of the framework, where the most preferable sets of sensors are identified from many aspects, e.g. qualities, performance, form factor, cost, power consumption, availability, aesthetics, etc. The trade-offs needs to be resolved based on overall requirements for the prototyping or product.

### 3.6 An Example to Use the Framework

Figure 3 illustrates a part of the framework utilization, that represents the selection flow for a person's presence on an office chair, i.e. sitting on a chair. In SETP1,



**Fig. 3.** An example of the framework usage (for person’s presence on an office chair)

it begins with the state-of-use of interest, i.e. “presence”, which can be divided into two relationship: 1) “putting” his/her hip on the seat and 2) “touching” his/her back to the back seat in STEP2. Then, in STEP3, five types of phenomena are identified, and then in STEP4, seven types of sensors are extracted based on the phenomena. Finally, in STEP5, a photo sensor and force sensor are selected for “hip-on-the-seat” detection, while a force sensor and touch sensor are chosen for “back-touching detection”. To enhance the reliability of the sensing, more than two same sensors are to be utilized. A temperature sensor, mechanical switch, microphone, and IR (infra-red)/US (ultrasonic) sensors are rejected due to the ease of broken by human weight, low response speed, and/or privacy violation.

Our proposal is not the entire structure of a selection flow, instead the five steps that systematically identify sensors for the state-of-use extraction. The selection flow should be improved by the evaluation after prototyping, and also it should be extended incrementally through the development. Sharing the selection flow with others allows an artefact developer to follow the same way as successful development so that he/she becomes free from the ad-hoc selection.

## 4 Discussion

### 4.1 Information Beyond State-of-Use

Sentient artefacts provide various kinds of information that is used as contextual information beyond the initial state-of-use as described below.

**Identity of Interacting Entity:** An artefact interacts with a human (user) and another artefact. In case of the interaction with a human, a personal artefact identifies a user with high accuracy because it is assumed that the user is equals to the owner. So, the state-of-use is a trigger for a personalized service like the

AwareMirror system [4]. An artefact also interacts with other artefacts, e.g. a cup on a desk. If the desk has an RFID reader, it can identify what is on it.

**Location of a User:** A stationary artefact like a door, mirror stand, etc. is the one that does not move. The characteristic allows a system to infer a user's location that offer the context with additional meaning. For example, a user in front of a mirror in the entrance might be in the different situation from who stands in front of a mirror in the washroom. The user's location is determined without any precise location sensing system if the location of the sentient artefact is known in some ways, e.g. manually specified.

**Adverbial Information of State-of-Use:** Adverbial context represents a user's context more richly, which can separate into two types: 1) an absolute and 2) relative adverbial context. Adverbial context with absolute values does not depend on a situation like a user's feeling and an application's requirements. In the statement "a user is sitting on a chair *with his/her back touching on the chair's back once per minute*", the italic part is the adverbial expression while "10 times per second" represents the user's situation of sitting absolutely. So, an application developer flexibly interprets it based on application requirements. However, an issue here is the selection of appropriate metrics, e.g. the frequency, average value, etc. On the other hand, the relative expression like "slowly", "suddenly", "angrily" is difficult to model at the development because such an expression is subjective. This is important because a sentient artefact should be a generic component for applications rather than an application specific one.

## 4.2 Spatially Distributed Artefacts Integration

A single sentient artefact provides a piece of a user's context. However, it is insufficient to describe more complex context in a robust way. So, sentient artefacts should communicate with external entities.

**Communication Styles:** The communication between an artefact and application is achieved in three ways: 1) event notification, 2) periodic transmission, and 3) retrieval from outside. Figure 4 illustrates the relationship among state, event, and also adverbial information. An event is generated as a result of an action performed by a user, e.g. sit down and stand up, while a state is represented by a period between events. In the figure, the state "sitting" represents from the event "sit down" to the other one "stand up". Adverbial information is attached to both an event and state (the rounded rectangle in the figure).

The event notification style is popular because many context-aware applications adapt their behavior according to the change of context. Also, it can minimize the communication. The periodic transmission is appropriate for sending monitored states continuously. However, this kind of communication should be minimized because it wastes the power of the artefact as well as bandwidth. The transmission should be invoked after detecting the start of original functionality of an artefact, e.g. "sitting", in order to avoid useless communication. This

is an important aspect for an everyday artefact that has long idle time. States can also be provided by retrieval from outside entity, where the entity can get information at anytime. The direction of the back seat and the user’s leg state, e.g. the left foot on the right, have been obtained in this way.

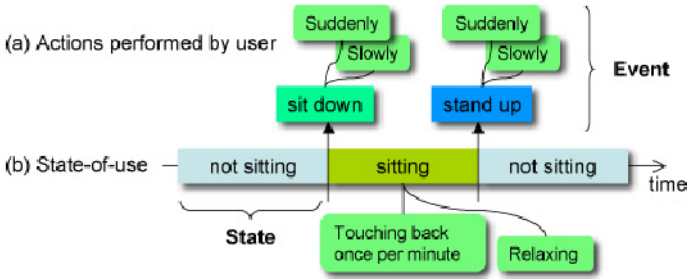


Fig. 4. Relationship among State, Event, and Adverbial Information

**Centralized vs. Decentralized Architecture:** We have developed an artefact integration middleware, *Bazaar*, which manages a model of world and provides a unified access to it[5]. *Bazaar* employs a centralized server architecture, where each artefact simply communicates with a server and application logics run. On the other hand, a decentralized peer-to-peer architecture has no central server and seems to be suitable for the artefacts communication because of scalability in the number of artefacts and the absence of single point of failure.

However, the application logic requires extraction of higher level context from each state-of-use. Also, it controls the presentation of context-awareness. We consider it is hard for a sentient artefact to execute these processes on behalf of others because it usually has limited resources. We expect a personal devices like a cellular phone and PDA to be a central server because they have enough resources. A user of the device is assumed to be its owner, which can identify the user and allows the device to provide a personalized service easily. Moreover, due to its local processing, privacy sensitive information does not reveal at all.

### 4.3 Towards Robust and Portable Application

A type of sentient artefact can vary in the level of abstraction that it perceives. As can be seen in Fig. 2, the sentient chairs shows various level of information abstraction, i.e. just presence of an object, and the way of sitting. The one with electronic compass also provides the direction of the seat back, a type of usage “facing”, in addition to the original one, “sitting”. If an application requires information of just presence of an object, all the three types of chairs can be utilized. This means these chairs are replaceable with each other at the place, and that the application running at one space is portable to another. Therefore, the state-of-use of chairs should be properly structured, that is, an ontology needs

to be specified. Moreover, an ontology for much wider class like an “artefact” than a single class “chair” allows an application to replace a chair with a door, for example, as an artefact that provides information of the user's presence.

Furthermore, to infer a user's context accurately and thus to build robust applications, the quality of information like *freshness*, *accuracy*, and *confidence* should be considered as well as the information contents. We need to identify suitable ones and include them into a context model as meta-information, which allows an application developer to implement appropriate adaptation strategies.

#### 4.4 Privacy Concerns

One of our design principles is augmenting artefacts with low-level sensing technologies, e.g. accelerometer, infra-red range finder, etc. This is because the user's feelings of privacy violation. A video camera and microphone can be utilized to detect rich context like the user's activity, identity, location, and even emotion and intension without the user's explicit input. However, our user survey at the AwareMirror development has revealed the testees' obtrusive feelings of being watched even though they know the benefits of the method[4].

In terms of user identification, the testees pointed out that the merit of using a sentient artefact that is hardly shared with others is its implicitness. As described in Section 2.2, it is utilized to start showing personalized information on the mirror without a user's explicit interaction with the mirror. However, the possibilities of intentional or unintentional use by other persons were also pointed. We consider users know the trade-off between the relevance of information and the efforts they need to make to keep it secret. So, explicit identification method like fingerprint recognition should be accepted if the system provides a highly relevant information like credit card number. Furthermore, if the toothbrush is utilized in a closed reliable group like a family, these issues are suppressed. Therefore a system should separate the contents utilization, e.g. using the identity, from the acquisition method, e.g. sentient artefact-based, biometric-based, etc., for the method's flexible selection based on the application requirements.

## 5 Related Work

Regarding to the deployment of smart environments, Sentient Computing[1] proposed a location sensing system that utilizes the ultrasonic and radio frequency signal [8]. To measure the position of an object within a cubic inch, it requires dense ultrasonic transceivers, which is impossible to deploy and maintain without special cares. Moreover, the system provides only location context, which means a system's awareness of a user's context is limited. However, a sentient artefact provides its user's state-of-use as a primary context of the user, so the information source is closer to the user, which is considered to be more accurate and meaningful to him/her. And, as described in Section 4.1, it can provide various types of information. The more the type of artefacts increase, the more the kinds of extracted context increases. Therefore, neither precise nor dense location sensing system is required.

The MediaCups project [2] and its succeeding project of SmartITs[7] provide insights into the augmentation of artefacts with sensing and processing. The notion of artefacts computing composed of sensor augmented artefact provides a mean to obtaining human context implicitly, which has been greatly influenced the notion of sentient artefact. We are working on representing an artefact formally and integrating them systematically, which must be applicable to sentient artefacts based on the SmartITs platform.

## 6 Conclusions and Future Direction

A sentient artefact is designed to perceive its usage as a user's context. We have introduced three artefacts' prototyping experiences, and proposed a conceptual framework that systematically identifies appropriate sensors for specific state-of-use of an artefact. The following discussions are presented 1) the information expected through the state-of-use of sentient artefact, 2) communication styles and architectural comparison towards sentient artefacts integration, 3) needs for the ontology development and meta-information definition for robust and portable application, and 4) the sensing technologies to capture context considering privacy aspects.

The sentient artefact-based context acquisition is expected to require less precise or no location system, and to provide accurate information than just touching an artefact, however, we need to assess them through practical application developments and user evaluation. Also, we are working on developing an artefact ontology including meta-information.

## References

1. M. Addlesee, R. Curwen, S. Hodges, J. Newman, A. Ward, and A. Hopper. Implementing a Sentient Computing System. *IEEE Computer Society*, pages 50–56, Aug. 2001.
2. M. Beigl, H.-W. Gellersen, and A. Schmidt. MediaCups: Experience with Design and Use of Computer-Augmented Everyday Objects. *Computer Networks, Special Issue on Pervasive Computing*, 35(4):401–409, March 2001.
3. M. Beigl, A. Krohn, T. Zimmer, and C. Decker. Typical Sensors needed in Ubiquitous and Pervasive Computing. In *Proceedings of the First International Workshop on Networked Sensing Systems (INSS) 2004*, pages 153–158, June 2004.
4. K. Fujinami, F. Kawsar, and T. Nakajima. AwareMirror: A Personalized Display using a Mirror. In *Proceedings of International Conference on Pervasive Computing, Pervasive2005, LNCS 3468*, pages 315–332, May 2005.
5. K. Fujinami and T. Nakajima. Towards System Software for Physical Space Applications. In *Proceedings of ACM Symposium on Applied Computing(SAC) 2005*, pages 1613–1620, March 2005.
6. A. Schmidt and K. V. Laerhoven. How to build smart appliances. *IEEE Personal Communications*, pages 66 – 71, 2001.
7. The Smart-ITs project. The smart-its. URL: <<http://www.smart-its.org/>>.
8. A. Ward, A. Jones, and A. Hopper. A New Location Technique for the Active Office. *IEEE Personnel Communications*, 4(5):42–47, October 1997.



# A Multi-dimensional Model for Task Representation and Allocation in Intelligent Environments

Victor Zamudio, Vic Callaghan, and Jeannette Chin

University of Essex, Wivenhoe Park, Colchester CO43SQ, UK

{vmzamu, vic, jschin}@essex.ac.uk

<http://ieeg.essex.ac.uk>

**Abstract.** In the future, homes will have numerous intelligent communicating devices, and the user would like to configure and coordinate their actions. Appliances and people in intelligent environments will have some degree of mobility. If the user wants to go from one place to another, using the same community, the agent should be able to generalize the service, trying to build an equivalent collection of coordinating services. This ‘work in progress’ paper addresses this issue by proposing a multi-dimensional model that allows visualisation of devices, temporal relationships, mutual interdependencies and the environment dynamics. The model both offers a simplified means of visualising the task space and the interdependencies together with a means of reasoning about algorithmic solutions to task processing. The work is aimed at supporting research into Pervasive Home Environment Networks (PHEN) which is funded by the UK’s Department of Trade and Industry Next Wave Technologies and Markets programme.

## 1 Introduction

Over the last decade, the rapid expansion of the INTERNET has opened new possibilities for communication with mobile phones and PDAs being used on an increasing scale. More recently the possibilities have extended to using the internet to control everyday living and working environments. A particularly popular and useful application of this work is, to assist elderly people [1], which require live longer in their homes. These kinds of system use a form of monitoring them, to help people with basic issues such as reminders, reports and control of devices. The overlap between pervasive (or ubiquitous) computing and intelligent agents has spawned the emerging area of Ambient Intelligence(AmI). This is a new multidisciplinary paradigm, which includes architecture, electronics, robotics, machine learning, etc [2], which has given rise to numerous new problems.

In this paper we propose a framework to solve problems related to task allocation in intelligent environments; particularly the formation of communities of communicating networked devices. We introduce a formalism about temporal communities, and discuss the advantages of this approach.



**Fig. 1.** The iDorm2 test bed

## 2 Related Work

### 2.1 Multi-robot Task Allocation and Cooperation

In robotics, although the problem of task allocation in multi-robots systems is well known, the efforts to formalize it are recent [3, 4, 5]. Cooperative robotics has become increasingly popular because it provides fault-tolerant and robust mechanisms to solve problems which a single robot would find difficult, if not impossible, to solve. In terms of fault tolerance, if one robot failed, other robots could continue with the task, albeit with a slightly degraded performance.

In Multi-Robots Task Allocation (MRTA) [3], a very frequent question is: which robot should execute which task? This simple question leads to more basic questions such as: what kind of task can the robots perform; what kind of task should they execute? These questions have been partially answered by other fields, such as operational research, combinatorial optimisation and set theory, amongst others.

Some of the core problems in MRTA relate to the heterogeneity of robots and tasks. In a multi-robot system, not all the robots are able to solve all the types of task that need to be accomplished. Gerkey and Mataric proposed a domain-independent taxonomy of MRTA problems [3] based on three axes: a) single-task robots (ST) vs. multi-task robots (MR), b) single-robot task (SR) vs. multi-robot task (MR), c) instantaneous assignment (IA) vs. time-extended assignment (TA). These three axes permit the description of a very wide spectrum of problems, abstracted in such a way as to aid the process of finding solutions..

Dudek et al [4] have proposed a taxonomy for robot collectives, using seven axes: collective size, communication range, communication topology, communication bandwidth, collective reconfigurability, processing ability and collective composition.

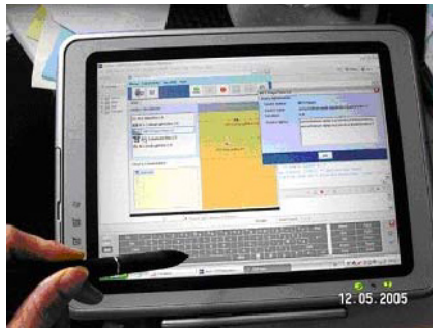
Chaimowicz et al [6] have proposed the use of a finite automaton approach, augmented with real-valued variables that changes with time. These hybrid automata can model continuous behaviour, communication and synchronization, and assume and exchange roles. The cooperation between several robots can be modelled by parallel automata.

## 2.2 Cooperative Groups in Intelligent Environments

There has been some work done relating to community formation in AmI environments. The Smart-its project [7], have developed a system that allows sensors, computational capabilities and communication to be added to artefacts. By embedding such systems into artefacts, logical groups of coordinating devices can be formed. For example, by adding load sensors to the corners of a table, they have been able to detect and track items on a table. Vildjiounaite et al [8], have created temporal sets of clothes (all the parts of a business suit), sets of ingredients for cooking a cake, or sets of items needed for travelling (passports, flight tickets, etc). Duman et al [9] introduced a system to autonomously learn the cause-effect association between the action of an agent and the devices connected to it which is used to identify and remove redundant connections.

Shahi et al [10], have developed the concept of a Personal Operating Space (POS), which permits the user to control and interact with the environment, using a smart phone, and OTIS (Object Transfer in Smart Spaces)[11], which provides adequate support to transfer PC sessions within spaces. Task Computing [12] developed in the Fujitsu Laboratories, allow the user perform complex tasks and create new services in ubiquitous environments. This research will provide the task processing engine for POS.

Task Oriented Programming (TOP), developed by Chin [13, 14], permits the creation of new “virtual appliances” or “communities of coordinating devices”, by establishing logical connections between the sub-functions of appliances (audio, video, etc.). This paradigm called “The Decomposed Appliance Model” permits a logical link between pervasive entities, thereby enabling them to coordinate actions creating so-called “virtual appliances”. For example, the telephone, the light and the TV could form a new community, where the TV could react when the telephone rings. This work will provide the task mobility processing for TOP. Figure 2 shows the System being used on a tablet.



**Fig. 2.** TOP system on a tablet

### 2.3 Learning and Prediction in Intelligent Environments

Prediction in intelligent environments has been done in several ways, from fuzzy rules to markov chain approaches. Doctor et al [15] proposed a paradigm that uses fuzzy rules to learn “on-line” user behaviour. The system learns the behaviour of the user and finds relationships between this behaviour and the devices the user interacts with seeking to pre-emptively set the environment to match the users expected needs at any particular time. This work is based on the axiom which might be surmised as “the user rules ok” meaning that at any time the user may override the agent by simply setting the environment to the state he needs (which are, in turn, reflected into the rules learnt by the system). Markov Decision Process has been used to predict the user’s next task. Panduranga et al [16] introduced the Task-based Markov Model (TMM), clustering the user’s behaviours as high-level tasks, and using a hidden Markov model to predict the next user action. These clusters of actions (which are the hidden states of the Markov model) are formed according: i) the time difference between successive actions, ii) the difference in the device location, or iii) the number of actions in the group.

## 3 The Challenge

As we mentioned before, both appliances and people using the home of the future will have some degree of mobility. A person using a several services at a time should be able to move from place to place and use an equivalent set of services; the system should be able to generalize the service, trying to build an equivalent collection of coordinating services. If we have a new device in the environment, the system should be able to incorporate it to at least one community. In general, the environment would contain redundancy, in the sense that there would be more than one device that could supply a service. The following scenario is offered to clarify these concepts:

### Part I

*Peter, after a busy day, arrives home. He goes to the master room, with a headache, because of the heavy traffic. So, he turns on only the indirect light. Then he configures the main TFT monitor with a movie about the ocean, with slow and tranquil waves, while the sun sets. He decides to listen to some quiet and relaxing music through the local speakers, and selects Air on the G string, by Bach. Besides that, he closes the only window blind. This environment (in technical terms a task or virtual appliance) is then saved by the system as on one of Peter’s personal preferences for future use (labelled by Peter as he as “Headache cure 1”)*

### Part II

*In this environment he relaxes, and because he feels hungry he begins to look for an Italian food receipt. He chooses the lasagne, and goes to the kitchen. When he is there, the preferences he expressed in “Part I” are translated to this new environment and the two windows blinds close leaving only the indirect light over*

*the table on. The music continues through the local speakers, and the familiar picture on the TFT monitor in front of the microwave now shows the video with the sea, and the monitor of the i-fridge shows the receipt. Great!*

### Part III

*The next week, his wife arrives home, really tired. She tells Peter she has a headache and asks him for a massage on her neck. Peter said: “Darling, this is better”, and activates the “Headache cure 1” task which turns on the indirect light and selects Air on the G string; then, the main TFT monitor shows a sunset in the beach, and the only window begin to close.*

In this scenario, some devices could be substituted (in the new environment) by a unique device: the speakers. Nevertheless, it is also possible that in the new environment that more than one device could perform the same (or equivalent) task. This is the case of the blinds, or the lights. The system should be able to choose which devices should be used to compose the new (equivalent) community, according to their location, user desires (preferences), or performance. Some complex configurations, such as a traditional TV, which is a device that includes several sub-devices (monitor, speakers, tuner, switches etc), will need every sub-function to be discovered and an equivalent community constructed. There are several problems arising from the scenario:

- formatting temporal communities: some devices could be performing temporal tasks.
- learning of communities.
- reconfiguring of communities.
- dynamic environments: devices will come and go from the network (eg due to purchase of new devices, failure of old devices or nomadic use).
- mobility of the user: the user could be moving to new environments, and asking for communities previously configured in other environments. This could be seen as a particular case of a dynamic environment.
- complexity of the devices: some devices could be performing more than one task at a time and there may not be one to one matches in functional elements (eg one to many or vice-versa).

## 4 A Multi-dimensional Model (MDM) of Pervasive Computing Space

We have developed a model of Pervasive Computing space that is formed using a 4D representation based on the following axis:

1. *Simple devices vs. complex devices.* A simple device can only perform one type of task, and can only perform one task at a time. Complex devices can perform several kinds of tasks at a time.
2. *Temporal tasks vs. non-temporal tasks.* A temporal task depends on time (eg are valid for a specific period). Non-temporal tasks do not depend on time.
3. *Coupled tasks vs. uncoupled tasks.* Coupled tasks have a mutual interdependency (ie are logically linked). Uncoupled task have no mutual dependency.

4. *Static vs. dynamic environment.* In a static environment, apart from system failure, devices do not move in time or space. In a dynamic environment devices come and go from the network.

In the next section we are going to formalize the problem, defining an allocation, a community, and an equivalent community. Then we will extend these communities in order to include time.

#### 4.1 Formalising the MDM Model-Allocations and Communities

An *allocation* is a duple  $(d, T)$  where  $d$  is a device and  $T$  is a not empty set of tasks, i.e.  $T = \{t_1, t_2, t_3, \dots, t_k\}$ , with  $k \geq 1$ . If  $k = 1$  we have a simple device, that is able to handle only one kind of task. This is the case of a speaker, or a microphone. If  $k > 1$  then  $d$  is a complex device, which is composed by other sub-devices, i.e. can handle more than one task. This could be the case of a TV, composed by a device that can handle two different kinds of signals: audio and video. When the user configures a new set of virtual appliances, he defines a new *community*. A *community*, denoted by  $C$ , is a finite not empty collection of  $n$  allocations, i.e.

$$C = \{(d_1, T_1), (d_2, T_2), (d_3, T_3), \dots, (d_n, T_n)\} \quad (1)$$

If the user goes to a new environment, the agent should create an *equivalent community*, denoted by  $C_{eq}$ . In order to create this *equivalent community*, for each allocation  $(d, T) \in C$  the agent should find an equivalent allocation  $(d_{eq}, T_{eq})$  in the new environment. As we mentioned before, we have two cases:  $k = 1$  and  $k > 1$ . i) If  $k = 1$  then  $d$  is a simple device and  $T = \{t_1\}$ . The agent should find a new allocation  $(d_{eq}, \{t_1\})$  such as the device  $d_{eq}$  is able to perform the only task  $t_1$ . ii) If  $k > 1$  then  $d$  is a complex device, and  $T = \{t_1, t_2, t_3, \dots, t_k\}$ . The agent should find, in the worst case,  $k$  allocations  $(d_{eq}^1, \{t_1\}), (d_{eq}^2, \{t_2\}), (d_{eq}^3, \{t_3\}), \dots, (d_{eq}^k, \{t_k\})$ , where every device  $d_{eq}^i$  is able to perform the task  $t_i$ , with  $1 \leq i \leq k$ .

#### 4.2 Formalising the MDM Model-Temporal Communities

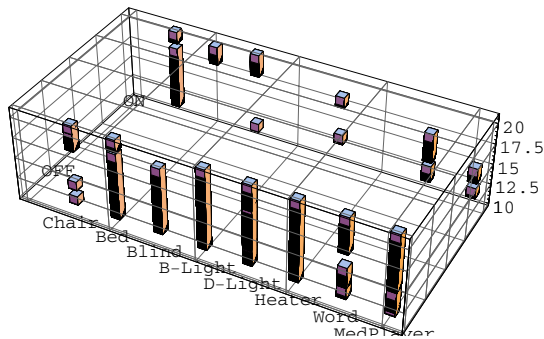
We could extend this framework in order to include time. A *temporal allocation* is a tuple  $(d, T, t_i, t_f)$  where  $d$  is a simple device,  $T$  is a (simple) task,  $t_i$  is the initial time and  $t_f$  is the final time. In other words, the device  $d$  will be performing the task  $T$  during  $t_f - t_i$  units of time, beginning on the instant  $t_i$ . So, a *temporal community*, denoted by  $C_t$  is a non-empty set of temporal allocations:

$$C_t = \bigcup_{j=1}^k (d_j, T_j, t_{ji}, t_{jf}) \quad (2)$$

As we mentioned before, some devices (with their tasks) could be coupled, in the sense that there is a logical link or causal dependency between them. This could be expressed in the following way: Let  $a = (d, T, t_i, t_f)$  and  $a' = (d', T', t'_i, t'_f)$  two different temporal allocations with  $t'_i > t_i$ . We say that  $a$  causes  $a'$  or in other words that  $a'$  is a consequence of  $a$  if every time that  $a$  occurs it implies that  $a'$  will occur. We will denote this by  $a \rightarrow a'$ .

## 5 Preliminary Results

We conducted a number of trials using a total of 18 users, in our test-bed the iDorm2 (see Fig. 1), in which they completed various tasks such as configuring and using communities. In the first of these there was a fixed configuration of eight devices; occupancy of a bed and sofa, status of the window blind, bed light, desk light, heater, telephone and a media player software application. During this trial we collected binary information on the status of devices in the environment and used this to create the visualisation shown in Fig. 3. This provides a graphical representation of the temporal community, with eight devices showing binary status *on* and *off*, and its evolution over time. This graphical representation of



**Fig. 3.** Representation of a temporal community

a community summarizes all the information related to time, task and devices. The advantages of this approach are that it provides a way of simplifying the visualisation of complex tasks (depending on the users focus, any of the 3 planes can be used to view and reason about the tasks). Thus the user interface could switch between these various views of the task space. In addition, this model maps directly to the underlying formalism used by the machine processes.

In the second trial we used the TOP system [13] which allowed the users to form and operate their own communities from a selection of 5 networked devices; smart sofa, two table lights, media player and telephone. Using information collected from the TOP system, we were able to model the user formation of communities of devices, expressing their cause-effect relationship.

Figure 4 provides a visual representation of the communities created by one of the TOP users. In this case, the sub-set of devices involved are the sofa, the desk-light and the MediaPlayer. There are several cause-effects relationships (or, in terms of TOP, rules). The first relationship is: when the sofa is *off* (ie, when nobody is sat on it), the desk-light and the MediaPlayer are *off* as well. The second relationship is: when the sofa is *on* (ie, when somebody is sat on it) the desk light and the MediaPlayer should be *on*.

## 6 Discussion

Our Multi-Dimensional Model (MDM) is able to represent the user interaction with the environment (the iDorm2); in particular, we were able to represent in a graphical way temporal communities of devices with binary status (*on* and *off*). We are working to extend the model in order to include continues values. The TOP system used to collect information let the user include *if-then* rules for the devices. MDM include this cause-effect relationships as shown in Fig. 4. At the moment, our system is addressing only the case when one antecedent could cause several consequences (as shown in Fig. 4), although Chin's TOP tool is considerably more powerful as it allows the user to create multiple antecedent and consequences. In due course we hope to consider these more complex cases.

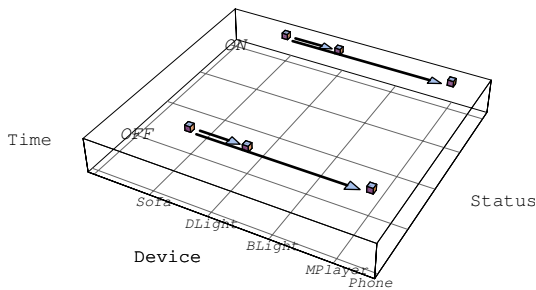


Fig. 4. A temporal community with causal relationships

## 7 Summary and Future Work

This paper describes ongoing work into task processing. The main contribution of this work is the Multi-Dimensional Model (MDM) for solving the problem of configuration and automated retrieving of communities. In particular, we have identified four main axis that should be taken into account in order to solve the problem of reconfiguration of communities. This model allows the user to be presented with differing views of the task spaces, simplifying his visualisation and understanding of the space together with enabling machine processing such as pattern matching schemes to be employed. Part of our longer term aims are to use the visualisation and formalisms we have presented to allow systems such as TOP, to reason about resource constraints, functional conflicts and mobility of mapping in system of coordinating pervasive computing devices. This approach has the following advantages:

- It lets the user interact with the system, with intuitive (and easy to remember) information, such as device-task, device-time, or even just single information, such as device or task.
- The system is fault tolerant. If the user does not remember exactly the community, the system can deal with the differences or erroneous directions generated by the user.



- This representation simplifies the problems related to dealing with the complexity of the devices, temporality of the tasks, and dynamics of the environments.
- The visual representation of the cause-effect relationships between allocations provide rules of evolution of the communities.
- The formalisms open up a way to reason about multiple tasks and their interaction.

Currently we are working to include continuous states in the representation of the devices, and to include temporality in the description of the cause-effect relationships. At the moment we are only considering a *one-to-many* relationships for firing rules, but in the future we will include *many-to-one* relationships (several causes for one consequence). The work described in this paper is ongoing. It builds on existing work at Essex University, in particular it is aimed at providing task processing support needed to underpin paradigms such as Chin’s Task-Oriented-Programming (TOP). At this stage we have established a representational and analytical model. Currently we are developing various task processing algorithms to use this model. For example, we are working to use pattern recognition techniques for retrieving the working communities. Also, we are investigating visual representations of the real time evolution of communities including cause-effect relationships. We look forward to reporting on these results at later conferences.

## Acknowledgment

We are pleased to acknowledge the UK’s DTI Next Wave Technologies and Markets Programme which has provided the underlying scientific challenge, access to state-of-the-art tools such as TOP and support for reporting this work. Victor Zamudio would also like to acknowledge the support of the Mexican National Council for Science and Technology (CONACYT). In addition, we would like to thank to Anuroop Shahi for his friendship and insight to the challenges of task programming that he provided. Finally we would like to thank Fernando Rivera-Illingworth for providing part of the experimental data set employed in this research.

## References

1. Haigh, K. Z., Kiff, L. M., Myers, J., Guralnik, V., Geib, C. W., Phelps, J., and Wagner, T.: “The Independent LifeStyle Assistant<sup>TM</sup>(I.L.S.A.): AI Lessons Learned”. In The Sixteenth Innovative Applications of Artificial Intelligence Conference (IAAI-04), July 25-29, 2004 San Jose, CA. Pages 852–857
2. Remagnino, P., Foresti, G. L.: Ambient Intelligence: A New Multidisciplinary Paradigm, Systems, Man and Cybernetics, Part A, IEEE Transactions on. Publication Date: Jan. 2005. Volume: 35, Issue: 1
3. Gerkey, B. P., and Matarić, M. J.: A Formal Analysis and Taxonomy of Task Allocation in Multi-Robot Systems, International Journal of Robotics Research, 23(9): 934-954, September 2004

4. Dudek, G., Jenkin, M., and Milios, E.: A Taxonomy for Multi-Agent Robotics. Robot Teams: From Diversity to Polymorphism, T. Balch and L. E. Parker (Eds.), 2002
5. Gerkey, B. P., Mataric, M. J.: A Framework for Studying Multi-Robot Task Allocation. In Multi-Robot Systems: From Swarms to Intelligent Automata, Volume II. A.C. Schultz and other (eds.), pages 15-26, the Netherlands, 2003. Kluwer Academic Publishers
6. Chaimowicz, L., Kumar, V. and Campos M. F. M.: "A Paradigm for Dynamic Coordination of Multiple Robots", *Autonomous Robots* 17(1): 7-21, July 2004
7. Holmquist, L. E., Gellersen, H. W., Kortuem, G., Schmidt, A., Strohbach, M., Antifakos, S., Michahelles, F., Schiele, B., Beigl, M., Maze, R.: Building Intelligent Environments with Smart-Its. *Computer Graphics and Applications*, IEEE Volume 24, Issue 1, Jan-Feb 2004 Page(s):56 - 64
8. Vildjiounaite, E., Malm, E., Kaartinen, J., Alahuhta, P.: Networking of Smart Things in a Smart Home. UBIHCISYS 2003 Online Proceedings. UbiCom 2003, Workshop 7. 2003. <http://ubihcisys.stanford.edu/online-proceedings/index.html>
9. Duman, H., Hagaras, H., Callaghan, V.: A Soft-Computing based Approach to Intelligent Association in Agent-Based Ambient-Intelligence Environments. Published at 4th. International Conference on Recent Advances in Soft Computing 2002 RASC2002. Nottingham, U.K. December 2002.
10. Shahi, A., Callaghan, V., Gardner, M.: Introducing Personal Operating Spaces for Ubiquitous Computing Environments. *Pervasive Mobile Interaction Devices 2005 (PERMID 2005)*, hosted by 3rd International Conference on Pervasive Computing, Munich 8-13, May, 2005
11. Shahi, A., Gardner, M., Callaghan, V.: Supporting Mobile Sessions Across Pervasive Smart Space Environments. The IEEE International Workshop on Intelligent Environments. University of Essex, 28th-29th June, 2005
12. Masuoka, R., Labrou, Y., Song, Z.: Semantic Web and Ubiquitous Computing - Task Computing as an Example - AIS SIGSEMIS Bulletin 1(3) October 2004
13. Chin, J., Callaghan, V., Hagaras, H., Colley, M., Clarke, G.: "End-User Programming in Pervasive Computing Environments", The 2005 International Conference on Pervasive Systems and Computing, Las Vegas, Nevada, USA, June 27-30, 2005
14. Callaghan, V., Colley, M., Hagaras, H., Chin, J., Doctor, F., Clarke, G.: Programming iSpaces: A Tale of Two Paradigms, Chap 24 in book iSpaces published by Springer-Verlag, June 2005
15. Doctor, F., Hagaras, H., Callaghan, V., Lopez, A.: An Adaptive Fuzzy Learning Mechanism for Intelligent Agents in Ubiquitous Computing Environments. Proceedings of the 2004 World Automation Conference, Seville, Spain
16. Rao, S., Cook, D. J.: Predicting Inhabitant Actions Using Action and Task Models with Application to Smart Homes, *International Journal of Artificial Intelligence Tools*, 13(1), pages 81-100, 2004

# Norms Enforcement as a Coordination Strategy in Ubiquitous Environments

Ismail Khalil Ibrahim, Reinhard Kronsteiner, and Gabriele Kotsis

Institute of Telecooperation, Johannes Kepler University Linz,  
Altenberger Str. 69, A-4040 Linz, Austria  
Tel.: +43-73224689888, Fax: +43-73224689829  
{ismail, reinhard, gk}@tk.uni-linz.ac.at

**Abstract.** Ubiquitous environments are characterized by their openness, dynamicity and autonomy. Electronic institutions are the agents' counterpart of human organizations, which are specifically designed for providing support, trust, and legitimacy in ubiquitous applications. In this paper, we propose an algorithm for norms enforcement in electronic institutions as a coordination strategy in ubiquitous environments by introducing substitution rules that map the norms set for the electronic institution to normative rules that can be implemented in the protocols to govern agents' interactions in ubiquitous environments.

## 1 Introduction

In ubiquitous environments [11], a set of mobile (mostly autonomous) entities, each with limited resources and knowledge, needs to interact with each other and users to achieve a common goal within a specific context. This interaction is generally characterized by 1) openness, describing the possibilities that these mobile entities can join or leave the environment, 2) flexibility indicating the degree the mobile entities are restricted by the environment rules and norms, 3) stability defining the predictability of the consequences of actions and 4) trustfulness specifying the extent to which users may trust the environment

The infrastructure that supports this ubiquitous environment must meet two requirements, first the internal autonomy requirement of participating entities in which interaction and structure of the environment must be represented independently from the internal design of the participating entities and second the collaboration autonomy requirement in which activities and interaction in the environment must be specified without completely fixing in advance the interaction structure.

In order to achieve these requirements, we argue that open ubiquitous environments can be modeled as electronic institutions where a vast amount of heterogeneous software and human agents interact in order to satisfy common and/or individual goals.

Electronic Institutions are the agents' counterpart of human organizations, which are specifically designed for providing support, trust, and legitimacy in agent mediated interactions [2][3][5][7]. They can be considered as computational frameworks that function the same way our human organizations function providing services and rules for software agents to meet in a way to create trust, prevent fraud

and reduce deception by verifying regulations which result in maintaining norms and insuring that all parties are conforming to these norms

Agents in this sense are computational entities that are situated in some environment and capable of flexible, autonomous action in that environment in order to meet its design objectives [12]. They can accept a task, negotiate or reject it. They should be capable to communicate, to coordinate, and to cooperate with other agents in order to solve a problem.

Norms in the context of electronic institutions are expectations about what behaviour, thoughts, or feelings are appropriate within a given group (of agents) within a given context (environment).

The study and modelling of norms in electronic institutions has attracted the interest of researchers from different disciplines such as sociology, economics, philosophy, and computer science.

Two approaches have been advocated for the design and modelling of norms in electronic institutions: in the cooperative normative behaviour [1][3][5], norms are defined implicitly through the behaviour of the agents and depends on how agents function in the institution; in the coordination strategy [10], norms are defined explicitly and the agents in the institution have to comply to these norms.

Most of the cooperative normative behaviour model is based on the assumption that different agents in the institution have some common goal or intention and this common goal enforces some type of cooperative behaviour on all agents in the institution. This means that the rules and regulations to which agents adapt their behaviour are hard wired in the protocols used by the agents to interact with each other. In the coordination model, actions of agents in the electronic institution are either rational or norm guided. In this case, norms are nothing but instruments of individual, collective, or generic optimization and to accept norms, as a motivational mechanism is not to violate methodological individualism, or to deny the importance of rational choice. In this case, the outcome of agent actions is a compromise between what the norm prescribes and what rationality dictates.

In this paper, we address the problem of norm enforcement in ubiquitous environments. We propose an algorithm for norm compliance checking by introducing substitution rules that map the norms set for the electronic institution as values to normative rules that can be implemented in the protocols to govern agents' interactions.

## 2 Norms as a Coordination Strategy

One of the most important challenges in open environments like ubiquitous computing, is the specification of mechanisms through which perspective participants can evaluate the characteristics and objectives of the environment rules in order to decide about participation.

Currently, in ubiquitous applications, agents are simply designed from scratch so that their behaviour complies with the behaviour described by the role(s) it will take up in the environment. Comprehensive solutions for this point require complex agents that are able to reason about their own objectives and desires and thus decide and negotiate their participation. As a first step for this is the comparison between the specification of agents and roles to determine whether an agent is suitable to enforce norms [2].

An important aspect concerning norm enforcement is that of modifying the agent internal architecture or beliefs to include the characteristics of the assumed role(s). A

possible solution for this has been proposed by [4], which extends agents with an interface to the society. This interface prevents any action not allowed by the role definition. However, it does not ensure the proactive behaviour expected from the role and is not flexible enough to incorporate different enacting styles, capabilities and requirements of the agents. It actually makes the actual agent “invisible” to the society and only its enactment of the role behaviour apparent in the society. We think that the consequence of an agent adopting a role is more drastic than this. The actual agent behaviour must often be modified according to the goals, norms and rights specified by the role.

In the following, we assume that agents have goals of their own, and are able to form (either by design or deliberation) plans to achieve those goals. These assumptions are consistent with the agent view and can be seen as a minimal agent definition. Here we describe agent societies from a global perspective, rather than from the perspective of the individual agents. Even though agents will take many roles simultaneously and along their life cycles, from the perspective of the society, each agent is a different individual. From our perspective, it is up to the agent how to manage and prioritize its goals. That is by assuming a role; the agent will receive the objectives from that role. How the agent will handle those objectives, whether it interprets them as goals or as norms, what priority it gives them, is up to the agent itself. However, the society model is based on the assumption that agents that take up roles are expected to eventually realize the assumed objectives.

Nevertheless, societies are concerned with judging the attitudes of the different areas and how those will affect the performance of the role. That is, the society will not look at the agent as a whole but at how a certain area is acting.

Agent literature discusses extensively different types of social attitudes of agents: selfish, altruistic, honest, and dishonest.

Different types of agents result in different role performances, because the way an agent will plan its goals, which is dependent on his social attitude, influences the realization of its role objectives and the fulfillment of the role norms. For instance, some types of agents will only attempt to achieve the goals of their adopted roles and forget their own private goals, while others will only attempt to achieve the goals from the role after all their own goals have been satisfied. Furthermore, the relations between agent plans and role objectives, and of agent goals and sub objectives must be considered

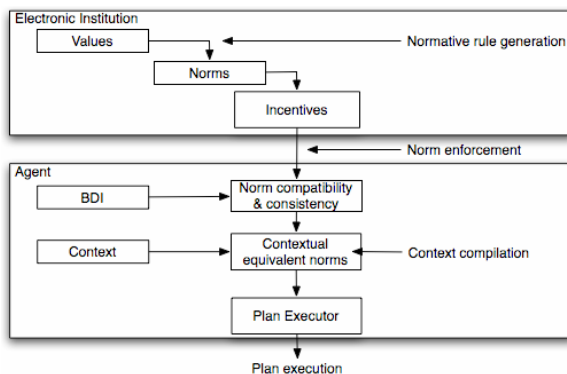


Fig. 1. Architecture of norm enforcement

In our approach [8], we suppose that we have a set of value definitions stored in the normative structure of a certain electronic institution. We consider the problem of given a norm  $N$  that is captured from the normative structure of the institution through semantics, how to find a rewriting  $N'$  for it using one or more of the available normative rules such that  $N$  is contained in  $N'$  (i.e., the value of  $N$  is a subset of  $N'$ ). It is well known that containment implies equivalence.

We consider the problem of verifying partial compliance and verifying complete compliance of agents to norms set for the electronic institution.

A partial compliance of a norm  $N$  using a set of normative rules  $R$  is a conjunctive norm  $N'$  whose head is a norm literal, such that  $N' \subseteq N$  and the body of  $N'$  contains one or more normative rule literals defined in  $R$  (it may contain value literals as well).

A complete compliance of  $N$  using  $R$  is a partial compliance of  $N$  using  $R$  whose body contains normative rules and built in predicates only (it may not contain value literals).

In our approach to norm compliance, we refer to the architecture depicted in figure 1, in which four phases are identified:

1. The normative rules generation phase where *values*  $V$  are expressed in terms of the normative rules  $R$ . The result of this phase is a set of normative rules  $R$  that contain all the necessary information for norm enforcement phase.
2. In the context compilation phase, the context [7] is used to compute a set of contextually equivalent norms ( $CEN$ ). These contextually equivalent norms  $CEN$  are filtered and stored for later use.
3. The norm compliance phase, which uses the contextually equivalent norms to decide whether, norm  $N$  is consistent to the set of norms defined for the electronic institution. This phase uses the set of normative rules generated earlier to rewrite the norm in such a way to generate an equivalent norm  $N'$  that produce the same effect as the original one but defined over the component normative rules.
4. The plan execution phase, where a norm  $N$  defined over the values is verified to comply to the electronic institution regulations and rules

## 2.1 Example Scenario (Adapted from [9])

In order to understand the system architecture, we will model the mobile auction scenario (mobile flea market) we presented in [9] as an institution. This example illustrates a number of important characteristics that is quite common to ubiquitous environments and must be taken into account in the development of an agent based computational market [9]. There are multiple services available from a number of agents representing independent participants. Multiple agents offer broadly similar services. The services themselves are described by multiple attributes (e.g. price, quality and delivery time). The services available change over time (new services may become available, or agents may alter the way in which existing services are offered). Services may differ in terms of the number and heterogeneity of the tasks involved in the delivery of the service and in their degree of independence and the type and frequency of interactions between different customers changes while the service is being delivered.

For the sake of clarification, let us assume that the mobile flea market sets the following vision as a value,

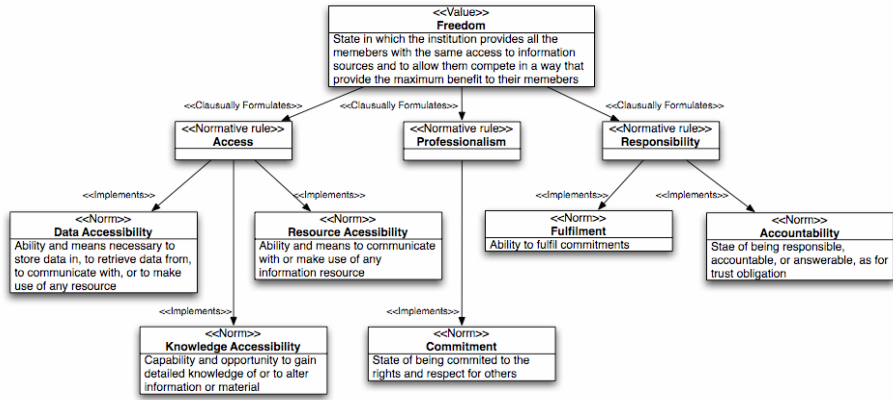


Fig. 2. Normative structure of the example scenario

*“Fostering the highest standard of free competition amongst those for whom the institution is responsible”*

Stakeholders in the institution are sellers, buyers, auctioneers, market owners, and accountants. The objective of the market owners is to provide a fair set up of the market where sellers sell their goods at the highest price, buyers find what they are looking for in an efficient way, and buy products with the cheapest price. All expect to have a win –win situation.

As we see, the notion of free competition is a very general concept and can be interpreted according to different stakeholder interests and intentions. This means that the value of fair competition can be translated into three normative rules; one is related to the access to resources within the institution; and the second is concerned with the professionalism with which the institution is operating; and the third is responsible management. These three normative rules can be further translated into low-level norms. The normative structure of the example is depicted in Figure 2.

Let us define two values for our conference institution:

*Accountability(Sellers, Buyers, Products)*

*Freedom(Access, Professionalism, Responsibility)*

Suppose the electronic institution maintains two normative rules:

Access(Data, Resources, Knowledge) ← Accountability (Seller, -, -, -)

Responsibility(Fulfillment, Commitment) ← Accountability (-, -, Buyers)

*And the following set of norms:*

- N<sub>1</sub>: A trading agent winning a bidding round within an auction is obliged to subsequently pay for the acquired good.
- N<sub>2</sub>: The seller agent is committed not to sell goods or products, which had been already sold.
- N<sub>3</sub>: The seller agent is responsible of delivering goods or products to the buying agents.
- N<sub>4</sub>: No bids after the deadline.

### 2.2 Normative Rules Generation

In this phase, the values are expressed in terms of the normative rules. The result is a set of rules that contains all the information needed for the norm enforcement phase. The generator can be thought as a specific model checker for normative rules.

Consider a normative rule of the form:

$$R \leftarrow V_1, \dots, V_n, C_1, \dots, C_m,$$

where

$V_1, \dots, V_n$  are value literals and

$C_1, \dots, C_m$  are constraints.

Let  $N_1, \dots, N_l$  be the distinguished variables and

$R_1, \dots, R_m$  be the rest of variables.

To avoid having to deal with the existentially quantified variables  $X_1, \dots, X_l$ , we skolemize the normative rule by substituting for all  $Y_j$  a different skolem term of the form  $f_j(N_1, \dots, N_l)$ . Let  $\theta$  be the substitution that skolemizes all such  $R_s$ .

Since all the normative rules predicates in  $R$  are completely defined by its body, then the normative rule implies the formula:

$$(\forall N_1, \dots, N_l) (R \rightarrow (\exists R_1, \dots, R_m) (V_1 \wedge \dots \wedge V_n)).$$

After skolemization, we derive  $n$  substitution rules below,

$$V_1\theta \leftarrow R$$

.....

$$V_n\theta \leftarrow R$$

where

$\theta = \{N_i \leftarrow f_i(N_1, \dots, N_l) \mid 1 \leq i \leq m\}$  is a substitution rule and  $f_1, \dots, f_m$  are  $l$ -ary skolem functions with arguments  $N_1, \dots, N_l$ .

For each substitution rule the head is a value literal, and every variable in the head also appears in the body.

The skolem function  $f_i(N_1, \dots, N_l)$  is a mapping or transformation from the normative rule to the value that assigns to each variable in  $R$  (i.e.,  $N_1, \dots, N_l$ ) a variable of  $V$  such that  $f_i(N_1, \dots, N_l) \in V$ . The normative rule is the domain of the skolem function and the value is the co-domain.

### 2.3 Norm Enforcement

Given a norm  $N$  defined over the set of values, a set  $R$  of normative rules  $\{R_1, \dots, R_k\}$  and a set  $\theta$  of substitution rules  $\{\theta_1, \dots, \theta_n\}$ .

Our goal is to find an equivalent norm  $N'$  that is compliant to the set of norms defined for the electronic institution. A norm  $N'$  of  $N$  using  $R$  can be obtained by rewriting  $N$  using  $R$  in such a way that the resulting norm contain one or more normative rules predicates defined in  $R$  and produce the same effect as the original norm for any given electronic institution.

Norm Enforcement process can be done in two steps:

The first step is to determine the valid norm substitutions. Norm substitution is the process of assigning to each value literal in the body of the norm the substitution rule(s)  $\{\theta_1, \dots, \theta_k\}$  that is defined in  $S$  for that literal. A substitution applied to the rule is the replacement of each variable in the rule by either a variable or a constant, while



a substitution applied to a norm is the replacement of each value literal by the substitution rules derived from the normative rules definitions for those literals.

The substitution is evaluated then for each variable in the head of the norm with respect to each substitution rule.

Proposition (1) determines when the substitution is valid and proposes the way to rewrite the norm.

**Proposition (1)**

*A substitution is valid if all the distinguished variables that appear in the head of the norm are substituted by the same variable in the substitution rule not by a skolem term*

A valid substitution means that the distinguished variable of the norm maps into the head of the normative rule that is defined by the substituted value and hence means that this distinguished variable comply to the normative rule as indicated by in proposition (2)

**Proposition (2)**

*For each  $n_i \subseteq V_i$  such that  $\forall i \theta_j \leftarrow R_j$*

$$\text{If } \sum_{\theta_j}^{V_i} n_i \text{ is a valid substitution for } V_i \text{ in } \theta_j \text{ Then } n_i \text{ comply to } R_j$$

Returning to proposition 1. The valid substitution for  $x_i$  means that  $n_i$  of the norm  $N$  maps into the head of the normative rule  $R_j$  defined by  $V_i$  and hence  $N$  complies partially to  $R_j$

Conversely, if  $\sum_{\theta_j}^{V_i} n_i$  is not a valid substitution, then  $n_i$  doesn't map into the head of

the normative rule  $R_j$  defined by  $V_i$  and hence there is no partial compliance for  $N$  in  $R_j$

The second step in the norm rewriting process is to construct the set of all the possible rewritings of the norm. This can be simplified by considering the set of all the substitution rules for each value in the body of the norm as an  $(m \times n)$  matrix where  $m$  is the number of the distinguished variables in the head of the norm and  $n$  is the number of the substitution rules defined in  $S$  for the value indicated by the matrix.

A valid substitution is considered to be 1 or TRUE and invalid substitution as 0 or FALSE to simplify the notation.

For example, if we have one value in the body of the norm, two substitution rules defined in  $S$  for that value, and three distinguished variables in the head.

So we construct the  $(3 \times 2)$  matrix for the norm as follows

$$\begin{matrix} & \sum_{\theta_1}^{V_1} n_1 & \sum_{\theta_2}^{V_1} n_2 & \sum_{\theta_3}^{V_1} n_3 \\ \sum_{\theta_1} & 1 & 1 & 0 \\ \sum_{\theta_2} & 1 & 0 & 1 \end{matrix}$$

Now to compute the possible rewritings of the norm described by this matrix, we need to evaluate the valid substitutions w.r.t each distinguished variable in the head of the norm.

**Proposition (3)**

A value literal  $V_i$  in the body of a norm  $N$  can be replaced by the normative rule literal  $R_j$  defined in  $R$  if

$$\sum_{\theta_j}^{V_i} n_1, \dots, n_n \text{ are valid substitutions for all the distinguished variables of the norm}$$

This means that if there is an all 1s row in our matrix, then all the distinguished variables are compliant to the normative rules and the value literal corresponding to that matrix can be replaced by the normative rule literal indicated by the row.

As suggested by proposition (3), any one of the normative rules can't replace the value literal in the norm since there is no all 1s row in its matrix.

Recalling proposition (2) where a valid substitution for any distinguished variable means that the normative rule defined in the substitution rule is partially compliant to the norm.

So we need to evaluate the partial compliance for the distinguished variables in the valid substitution matrix.

**Proposition (4)**

The conjunction of all the partial compliances for a norm is a rewriting if there exists

$$\sum_{\theta_j}^{V_i} n_i \text{ that is a valid substitution for all the distinguished variables in the}$$

substitution matrix.

For the norm in example 4 we have the following partial compliances:

- For  $n_1$ :  $R_1 \vee R_2$
- For  $n_2$ :  $R_1$
- For  $n_3$ :  $R_2$

Since there is partial compliance for all the distinguished variables in the norm. Then the conjunction of these partial compliances is a rewriting and we get the following rewriting

$$N(n_1, n_2, n_3) \leftarrow R_1(n_1, n_2, R_1, R_2), R_2(n_1, R_3, n_3, R_4)$$

For this norm we have the following two valid substitutions matrices:

$$\begin{matrix} \sum_{\theta_1}^{V_1} & \sum_{\theta_1}^{V_1} n_1 & \sum_{\theta_1}^{V_1} n_2 & \sum_{\theta_1}^{V_2} & \sum_{\theta_1}^{V_2} n_1 & \sum_{\theta_1}^{V_2} n_2 \\ \sum_{\theta_2} & 1 & 1 & \sum_{\theta_2} & 1 & 1 \\ \sum_{\theta_2} & 1 & 0 & \sum_{\theta_2} & 0 & 0 \end{matrix}$$

From the substitution matrices above we notice that  $R_1$  can replace each  $V_1, V_2$  in the norm so the norm can be rewritten as follows:

We also get the same rewriting from the partial compliances for this norm. For this norm we have the following two valid substitutions matrices:

$$\begin{array}{ccc}
 \sum_{\theta_1} & \sum_{V_1} n_1 & \sum_{V_1} n_2 \\
 \sum_{\theta_2} & 1 & 0
 \end{array}
 \quad \text{and} \quad
 \begin{array}{ccc}
 \sum_{\theta_1} & \sum_{V_2} n_1 & \sum_{V_2} n_2 \\
 \sum_{\theta_2} & 0 & 0
 \end{array}$$

Again we notice that  $R_1$  can replace  $V_1$  in the norm, while there is no normative rule that can replace  $V_2$ , hence the following is a rewriting of the norm.

$$N(n_1, n_2) \leftarrow R_1(n_1, n_2, R_2, R_3), V_1(n_1, \dots, n_3, R_4).$$

There is no rewriting that can be derived from the partial compliance since  $n_1$  and  $n_2$  have no valid substitutions in  $R_2$ .

### 3 Conclusion and Future Work

Agents are either always comply with norms or autonomously choose whether to do so or not based on the utility or situation.

This may cause conflicts between the electronic institution goals and the goals of agents within it. On the one hand, if agents are assumed to comply to norms, this may decrease the opportunities agents have to achieve their goals. On the other hands, if agents choose whether to comply to norms set by the institution or not, although agent goals may be satisfied, the electronic institution becomes unpredictable when all agents choose not to comply to norms and consequently result in violations of the overall essence of the organization

The purpose of the project from which this research paper stems is to build a complete monitoring system for electronic markets. Our objective is to integrate seamlessly the cooperation and coordination models as a basis for modelling electronic institutions.

### References

- [1] C. Castelfranchi, F. Dignum, C. Jonker, and J. Treur, "Deliberative Normative Agents: Principles and Architectures", *Proc. of the 6th Workshop on Agent Theories, Architectures and Languages, ATAL'99*, Orlando, 1999, pp. 206-220.
- [2] M. Dastani, V. Dignum and F. Dignum, "Organizations and Normative Agents", *Proceedings of first Eurasian Conference on Advances in Information and Communication Technology*, Teheran Iran, 2002, pp. 982-989
- [3] F. Dignum, "Abstract Norms and Electronic Institutions", *Proceedings of the Int. Workshop on Regulated Agent-based social Systems: Theories and Applications RASTA '02*, Bologna Italy, 2002
- [4] V. Dignum, *A Model for Organizational Interaction based on Agents, founded in Logic*, PhD thesis at Utrecht University 2004. SIKS Dissertation Series No 2004-1

- [5] M. Esteva, J. Padget and C. Sierra, "Formalizing a Language for Institutions and Norms", J.J.Ch. Meyer, and M. Tambe, (eds.) *Intelligent Agents VIII*, Vol. 2333 LNAI, Springer 2001
- [6] I.K. Ibrahim, E. Weippl, F. Dignum, and W. Schwinger, "Modelling Norms in Electronic Institutions" *Proc. of the Workshop on Agent-Based Simulation*, Passau, Germany 2002.
- [7] I.K. Ibrahim, G. Kotsis and W. Schwinger, "Mapping Abstraction of Norms in Electronic Institutions", *Proceedings of the 12<sup>th</sup> IEEE International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises WETICE'03*, Linz, 2003, pp. 30-35
- [8] I.K. Ibrahim, G. Kotsis and R. Kronsteiner, „Substitution rules of norm-compliance in electronic institutions“, *Proc. Of International workshop on enabling technologies: Infrastrucutre for collaborative enterprices*, Modena June 2004.
- [9] I.K. Ibrahim, R. Kronsteiner and G. Kotsis, "Agent-based mobile auctions: The flea market scenario", *Proc. Of 2004 Research conference in innovations in information technology (IIT2004)*, Dubai Oct. 2004.
- [10] F. López y López, M. Luck, M. and M. d'Inverno, "Constraining autonomy through norms" *Proc. of the 1st Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2002* ACM, 2002, pp. 674-681
- [11] J. Ma, L.T. Yang, B.O. Apduhan, R. Huang, L. Barolli and M. Takizawa, "Towards a smart world and ubiquitous intelligence: a walktrough from smart things to smart hyperspaces und UbicKids", *Journal of pervasive computing and communications I(1)*, Troubador publishing, March 2005.
- [12] J. Vázquez-Salceda and F. Dignum, „Modelling Electronic Organizations“, V. Marik, J. Mueller and M. Pechoucek, (eds.), *Proceedings of Central and East European Conference in Multi-Agent Systems CEEMAS'03*, Springer, Prague, 2003

# A Java-Based RFID Service Framework with Semantic Data Binding Between Real and Cyber Spaces

Kei Nakanishi, Makoto Setozaki, Jianhua Ma, and Runhe Huang

Faculty of Computer and Information Sciences, Hosei University, Tokyo 184-8584, Japan  
{nakanishi, setozaki}@malab.k.hosei.ac.jp  
{jianhua, rhuang}@k.hosei.ac.jp

**Abstract.** Recently there are getting more and more systems and applications using RFID. There is a necessity of a framework for them. This paper presents a Java-based framework that offers a set of general services to support various RFID systems for different purposes and application scenarios. The framework emphasizes on the semantic data binding for contextual information mapping between real and cyber spaces. The Java interface classes are provided to support general communications among a RFID, a reader and an application. Real and cyber spaces are synchronized via dynamic and real-time mapping from symbolic strings or IDs to the semantic XML data representations that are more effectively and efficiently useable by RFID applications. In this paper, the architecture and functional modules of the Java-based RFID service framework are explained in detail.

## 1 Introduction

The ubiquitous computing is aimed at supporting human's memory, judgment and action with pervasive computers and networks throughout physical environments [1]. RFID (Radio-Frequency-Identification) is enumerated as a technological element to achieve the ubiquitous computing. It is used as one of the means to acquire information in a real space. There are more and more studies on RFID related systems and developments, such as location-aware information technology [2], location tracking system [3], reminding system [4], human detection [5], etc. One of the main common features in developing an RFID-based system is the mapping of information on a real space to a cyber space.

There are two essential issues in developing RFID systems. The first is that a RFID operation method requires a library to operate RFID data and it is not easy to operate RFID data without a common interface API even if having the library in developing various RFID systems. It is usually a design requirement that the interface for reading and writing RFID data must be general though a communication means of reading and writing RFID data depends on its concrete type of RFID. It is also preferred that the interface and operation method can be used in different RFID systems by only switching the library. The second is the mapping between real and cyber spaces. A RFID keeps only symbolic code or ID data, i.e., a set of strings, that doesn't have much meaning if it is not associated with a real object or some real objects. Information in a real space in general has a special meaning, this framework is therefore

aimed at offering a general service to convert symbolic codes into semantic data so as to closely link real and cyber spaces.

In the rest of this paper, we first present several representative RFID based application scenarios, next summarize the common features of the RFID systems, and then propose and discuss a Java based framework. It consists of three basic parts: a RFID programming interface, the information mapping between real and cyber spaces using RFID and XML data binding, and intelligent application in a cyberspace. Finally, related work is discussed, conclusions are drawn, and future work is addressed.

## 2 Representative RFID System Scenarios

To capture common features of RFID systems, this section shows several RFID-based application scenarios: object finding, thing reminder, person location detection, and meeting member detection, respectively.

**Object Finding System.** This system is supposed to be able to find an object or some objects in a room or house where RFID tags with ID codes and readers are placed. Some RFID tags are put under the floors in a pre-designed layout. Such a tag is named *position tag* that keeps a symbolic number, which is associated with the coordinates of the tag stored in a database. In a room, a robot carrying a RFID reader and wireless LAN can move in a pre-defined course. The robot reads all RFID tags that are near the RFID reader. The obtained tag IDs are sent to the database via the wireless LAN and then used in a location analysis process as shown in Fig. 1. If a tag attached to an object is read, the four surrounding position tag’s coordinates determine the object’s position coordinate.

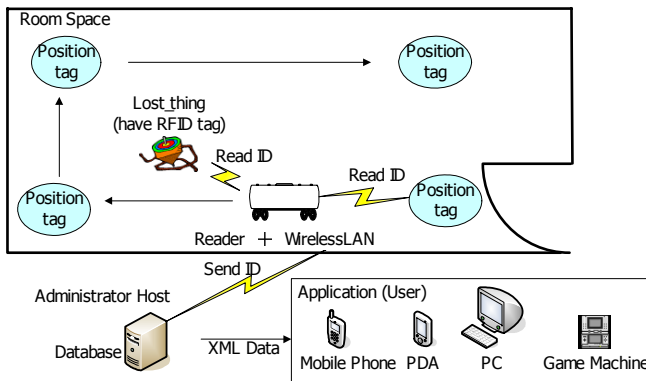


Fig. 1. A scenario of an object finding system

The process continues when the robot is moving and the obtained coordinate data is used in the application for drawing a map of the room on a GUI. All the related objects in the room are drawn on the map. To find a lost object, a user inputs the object name, and the system will inform the location of the object via some audio/visual interface.

**Thing Reminder System.** This system is supposed to give you advice messages in a room/house entrance when you forget to take something with you. This system requires that there is a RFID reader at an entrance, all things for a user to bring are attached RFID tags, and the item IDs are input to the application in advance. When the user goes out, the reader at the entrance reads the user’s identity tag and the items’ tags. These IDs are sent to a database via wireless LAN and then used in a process as shown in Fig. 2.

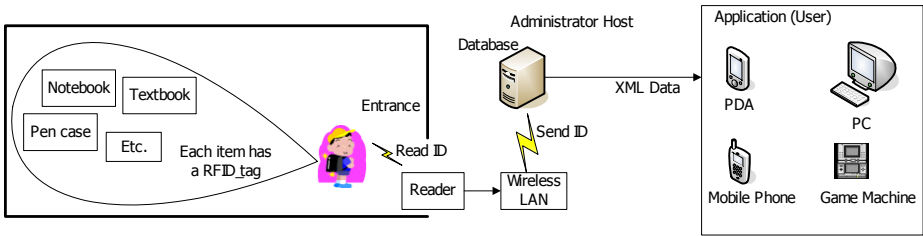


Fig. 2. A scenario of a thing reminder system

The obtained ID data is compared with the ID data in the database that the user input beforehand. If there is unmatched ID data that remains in the user input ID data set, the application reminds the user in some way such as a speech message of “you forget to take the desk pad”.

**Person Location Detection System.** This system can get a person’s location by detecting RFIDs carried with the person. The system requires at least two RFID tags per person and some RFID readers. Two tags are attached to each bottom of the person’s shoes. Some RFID readers are buried under the floor in special places such as an entrance, a kitchen, a toilet, etc. The location data in XML format of RFID readers is stored in a database. If the person steps over a RFID reader, the reader reads the person’s RFID tags and sends the IDs to the database via a wireless LAN that is connected with the reader. The tag IDs and RFID reader’s location are used to locate the person’s position in the application as shown in Fig. 3.

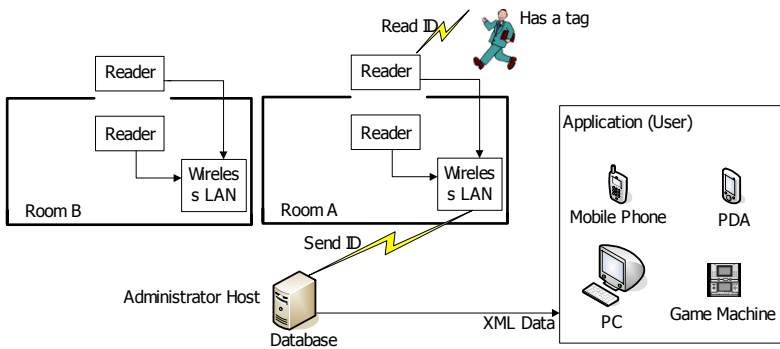


Fig. 3. A scenario of a person location detection system

In the application, there is a graphic image/icon that corresponds to a person’s tag ID. On the monitor, a house/building map is showed. Once a reader detects a person’s entering/leaving a room, an image/icon corresponding to the person’s tag ID is drawn in a certain position on the map. Watching the map, one can know who is in the room and where the room is in the house/building.

**Meeting Member Detection System.** This system is supposed to let a user know who is meeting whom by detecting RFID tags carried by the persons in a meeting. It requires that every one have a RFID tag and a reader with a wireless LAN. For example, a person A is going to meet person B and C. If A approaches to B, A’s reader reads B’s tag. After that, the tag’s data is sent to a database via the wireless LAN. In the database, the ID data is converted to XML-based data representation. In the application, a message generated from the XML data, such as “A is meeting with B”, is displayed on a monitor. If C is joining the meeting, with the similar processing, a message, such as “A is meeting with B and C”, is displayed on the monitors. A scenario of such a system is given in Fig. 4.

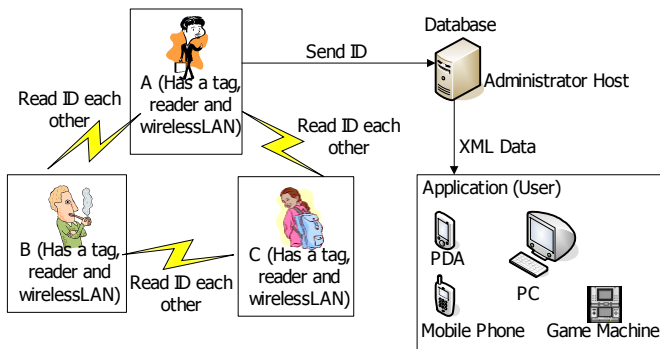


Fig. 4. A scenario of a meeting member detection system

### 3 RFID Service Framework

The four representative scenarios show some common features and process.

- A RFID reader reads a RFID tag.
- The ID is sent to a database.
- The database is used by an application.
- The application processes the current incoming data with comparisons of the data kept in the database.
- From the comparison results, the system can get conclusions, such as where is the object to be found, which object is forgotten, where a user is, who is meeting whom, or others.

As a result of the common features and the process, some commonly necessary components and their relations in a RFID system can be summarized in Fig. 5.



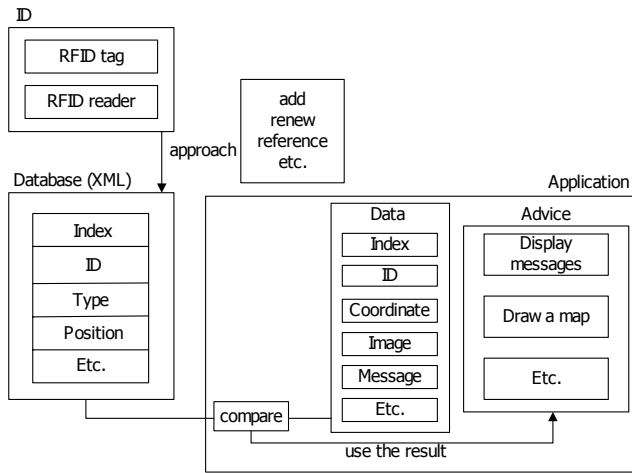


Fig. 5. Common components and a general structure of RFID systems

Further, the architecture of a general RFID service framework is resulted in to facilitate RFID service system developments and given in Fig. 6.

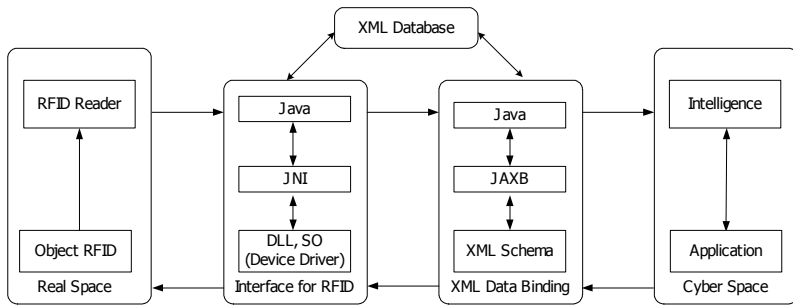


Fig. 6. A general RFID service framework

The RFID service framework is able to support a series of the processes. It mainly consists of two mechanisms, an interface API for RFIDs and XML data binding to bridge the real space and the cyber space. In the interface for RFID, the device drivers in DLL (dynamic link library) or SO (shared object) are necessary in operating specific RFID devices but can not directly be handled or accessed by or from Java. However, JNI (Java Native Interface) [6] can play the role of a bridge between a device driver in DLL or SO files and Java VM. Java VM can operate or access DLL and SO files through JNI. Thus, RFID service developers can easily operate RFID devices related data with using Java in an application layer. XML is used as a general data format in the system. To access and handle XML data in Java, JAXB (Java Architecture for XML Binding) [7] is imported to bind an XML schema to a representation in

Java code. As for application developments in Java, developers may only focus on how to use the data in their applications and do not need to know the details about how to get physical data and convert them into a correct format used in Java since these two steps are supported by this framework. Thus, a Java application processes information obtained from a real space as well as information on a cyber space.

## 4 Programming Interface for RFID

There are some common basic issues in the development of RFID-based systems. One is related to available methods of controlling the RFID reader and writer. The control methods for different kinds of RFID vary. Some kinds of RFID readers use RS-232 interface to connect to a computer. There is a case that a RFID is controlled directly by commands from neither the driver nor the library. Even if the driver is offered by a manufacturer, it is often impossible to control RFIDs at once because it is necessary to prepare the interface that connects a system and libraries. Next one is that RFIDs keep only symbolic ID codes. To operate the data from RFIDs, the codes should be closely related to their attached real objects. Another one is that a system should be general though a controller depends on kind of RFID used. Therefore, the mechanisms for accessing or operating any devices such as RFIDs and sensors to acquire information on a real space should be abstract. To enable such accesses to different kinds of RFIDs more easily, our framework adopts the XML technique. Though RFID has only symbolic number, it is possible to give each symbolic code a meaning and include it in a XML data file. Of course, for a XML file, it is impossible to record all data in a continuous time in a real space. Only those critical or turning points from which the changing trace of an object can be represented or recovered are recorded.

A parser is needed to read the XML data. Though the parser for different programming languages has been distributed by some ventures, most of APIs that can read/parse XML are offered in Java such as JAXM, JAXB and JAXP. They are included in Java XML Pack distributed by Sun Microsystems. Taking Java and XML's advantages of platform independence and object-oriented model, we use Java as implementation language of the interface for RFID in this research.

A driver and its associated library distributed by a manufacturer are often written in native codes. However, Java VM cannot run the native codes directly. To make them run-able by Java VM, one approach is to use JNI that is the API used by Java programs to invoke the native code compiled from C/C++ codes. The JNI can generate a Java header to use the native code in C/C++. Figure 7 (in left) shows the relation between the JNI and C/C++.

A library might be written in not only C/C++ but also C#. Like Java, C# is a programming language that runs on a special environment named CLR (Common Language Runtime). The CLR is a managed code environment for the .NET Framework [8]. When a program is written in a managed code that runs on CLR, even if JNI is used, it is not run-able by Java VM. Though Java VM can use native codes with JNI, it cannot use managed codes. However, if the managed codes are wrapped with the native codes, it is able to use them with JNI [9, 10]. Figure 7 (in right) shows the relation between the JNI and C#.

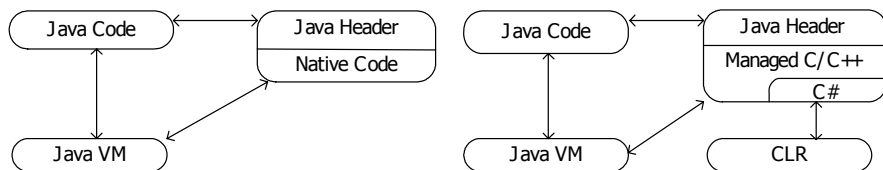


Fig. 7. The JNI as a bridge between Java and other languages

Thus, the interface based on Java is developed by using JNI, and a RFID ID code is read and send to the XML database. Its architecture is shown in Fig. 8. The interface is mainly to operate RFID. The next section explains how to map information from a real space to a cyber space with XML data binding.

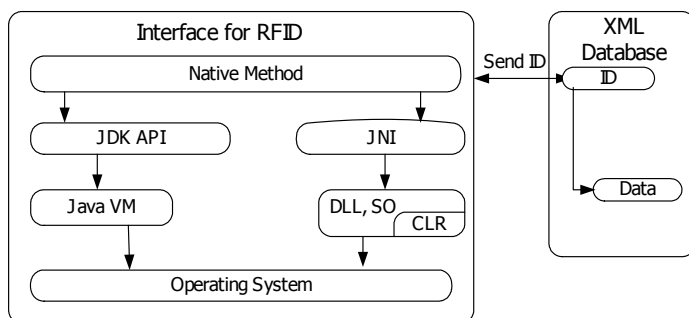


Fig. 8. The interface for RFID

## 5 Mapping Between Real and Cyber Spaces

Information obtained from a real space is about to be used by an application or some applications in a cyber space. However, information original from RFID readers is just a sequence of symbolic strings. Without attaching identical meanings, it is difficult to use such symbolic numbers by an application in a cyber space. To solve the problem, XML techniques play two important roles here. One is to use the XML scheme and attach some identical meaning to each symbolic code by adding necessary tags into the XML scheme. Another is to use the information represented in XML scheme by a Java application. In this case JAXB is required to parse XML structure and then generate a tree structure and further then map the tree into Java classes. Thus, information obtained from a real space is operational in a cyber space. The XML techniques link two kinds of spaces together and bind data from a real space to a cyber space, which makes two kinds of spaces reflective to each other. An example of the mapping between real and cyber spaces is shown in Fig. 9.

To operate a set of the data represented in XML in a cyber space is to make them being easily used by Java programs in a related RFID system or so called application. To different applications, data has different meanings. If an application is regarded as an object, data modeled in the XML scheme is a kind of object-oriented data. Both

Java and XML are based on object-oriented models. So the structure of XML can be easily mapped to the class of Java. An API that maps a XML model to a Java model is provided in the data-binding tool, JAXB, which enables to map the data obtained from RFID to XML and pass the XML to a cyber space. Consequently, it is achieved to map information from a real space to a cyber space.

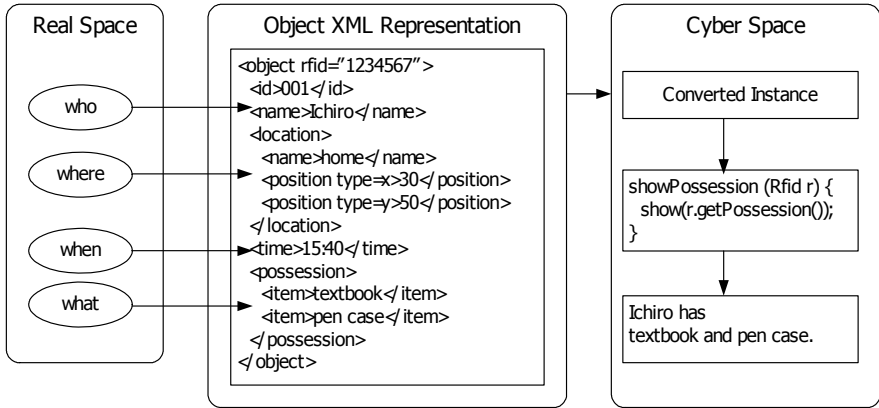


Fig. 9. An example of the mapping between real and cyber spaces

## 6 Intelligent Application in Cyber Space

A mechanism with which information on a real space is mapped to a cyber space is achieved by binding RFID and XML data. An application on a cyber space is a system that operates data acquired from a real space to offer some specified services. It is a fact that even if only critical data about objects from a real space is recorded, there is still much more data than it is necessary. To a specific application, how to dig out the data the application really needs is an important step. So called data mining or intelligent data processing is one of the important steps, which should be included as one part of the application. Besides, the application should have a specific algorithm or mechanism that uses or handles required data. For instance, in the application of finding a child's toy, the coordinate data is a kind of required data and can be obtained by such a RFID system. But how the location of the child toy is detected using this coordinate data is depended on an algorithm that the application adopts. The system does not limit the usage of the data.

It is worthwhile to emphasize that real and cyber spaces are synchronized by dynamically and real-time binding between RFID and XML data. With such synchronization between the two spaces, auto-login to a network might be possible. For instance, when a user moves from one room to another, the application changes the location data and the workgroup data in the XML at the same time. Even if a user does logout the current network, he/she can automatically login to the next network in another room as shown in Fig. 10.

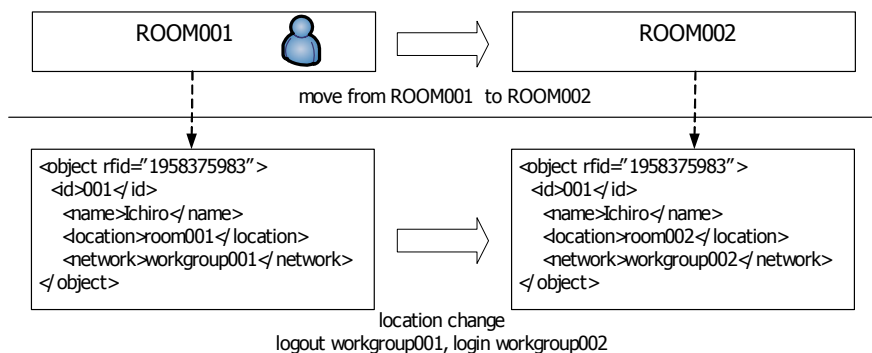


Fig. 10. The synchronization between a real space and a cyber space

## 7 Related Work

Mapping between real and cyber spaces by XML data binding is used in Robotic Communication Terminals [11]. The system focuses on barrier and barrier-free information on facilities at the National Institute of Information and Communications Technology. It converts object information on the real space obtained from the camera into XML, and does the mapping to a cyber space. Interactive Block System [12] also uses XML to map information between a real space and a cyber space. To support a variety of multimedia contents flexibly, the data is dynamically read. Compared to Robotic Communication Terminals and Interactive Block System, our research provides a common service framework for various RFID systems and Java applications. Semantic web and grid computing [13] use XML as metadata. The metadata technologies developed for the semantic web are applicable and relevant. XML data binding between real and cyber worlds needs to automate the management of the metadata. Our framework offers the service to manage the metadata by synchronizations between a real space and a cyberspace represented in XML. The applications greatly depend on situations and are hard to be developed one-by-one without a common framework. The Java Context Awareness Framework (JCAF) [14] is aimed at solving the problem that programmers develop context-aware applications.

## 8 Conclusions and Future Work

This research, as a part of UbiKids Project [15], is focused on making a general RFID service framework that assists the developments of various RFID based applications in which acquired information/data by RFIDs from a real space is used and processed. In the proposed framework, XML is used as a unified format to represent semantic data kept in a database that stores information about or from a real space. Such format makes the developments of Java based applications easier. Moreover, it can avoid the repeat work in developing RFID applications by providing common required services in any RFID based applications.

RFID is one kind of the important devices used to achieve the ubiquitous computing. Other devices such as the GPS, sensors, microphones, and cameras also play important roles in acquiring information from a real space. Moreover, these devices can be used to get other types of contextual information while RFIDs can keep only ID numbers. The interface that can control RFID as well as these devices is greatly in need and it will be added to our future work.

## References

- [1] S. Yamada and E. Kamioka "An Overview of Researches on Ubiquitous Computing Networks", *NII Journal No.5*, pp. 41-47, 2003.
- [2] Kenpei Morishima, Takahiro Konno and Syuhei Watanabe "Location Information Technology by RFID", *NTT COMWARE TECHNOLOGY*, pp.12-15, 2004.
- [3] K. Okuda, S. Yeh, C. Wu, K. Chang, and H. Chu, "The GETA Sandals: A Footprint Location Tracking System", Proceedings in *Location- and Context-Awareness*, May 2005.
- [4] G. Borriello, W. Brunette, M. Hall, C. Hartung and C. Tangney, "Reminding About Tagged Objects Using Passive RFIDs", Proceedings in *UbiComp 2004*, September 2004.
- [5] K. P.Fishkin, B. Jiang, M. Philipose and S. Roy, "I Sense a Disturbance in the Force: Unobtrusive Detection of Interactions with RFID-tagged Objects", Proceedings in *UbiComp 2004: Ubiquitous Computing*, pp. 269-282, September 2004.
- [6] Calvin Austin and Monica Pawlan, "JNI Technology", *Advanced Programming for the Java 2 Platform*, pp.207-230, November 1999.
- [7] JAXB, <http://java.sun.com/xml/jaxb/>.
- [8] MSDN .NET, <http://msdn.microsoft.com/netframework/>.
- [9] J. Bishop, R. N. Horspool, B. Worrall, "Experience in Integrating Java with C# and .NET", <http://www.cs.up.ac.za/cs/jbishop/Homepage/Pubs/Pubs2002/Bishop-Integrating-1.pdf>.
- [10] J. Bishop, R. N. Horspool, B. Worrall, "Experience with Integrating Java with New Technologies: C#, XML and Web Services", [www.cs.uvic.ca/~nigelh/Publications/ccpe03.pdf](http://www.cs.uvic.ca/~nigelh/Publications/ccpe03.pdf).
- [11] H. Fujiyoshi, M. Okada, T. Komura, I. Yairi, L. K. Kayama, H. Yoshimizu, "Robotic Communication Terminals as a Mobility Support System for Elderly and Disabled People", In *The 18<sup>th</sup> Annual Conf. of the Japanese Society for Artificial Intelligence*, 2004.
- [12] Y. Ito, Y. Kitamura, H. Kikuchi, K. Watanabe and K. Ito, "Development of Interactive Multimedia Contents Technology That Unites Real Space and Virtual Space in Seamless - Interactive Block System -", <http://www.ipa.go.jp/SPC/repo/rt/03fy-pro/mito/15-757d.pdf>.
- [13] Carole Goble and David De Roure, "Semantic Web and Grid Computing", in *Real World Semantic Web Applications, vol. 92, Frontiers in Artificial Intelligence and Applications, V. Kashyap and L. Shklar, Eds.: IOS Press, 2002*.
- [14] Jakob E. Bardram, "The Java Context Awareness Framework (JCAF) – A Service Infrastructure and Programming Framework for Context-Aware Applications", Proceedings in *Pervasive Computing*, pp.98-115, May 2005.
- [15] Jianhua Ma, L. T. Yang, B. O. Apduhan, R. Huang, L. Barolli, M. Takizawa, "Towards a Smart World and Ubiquitous Intelligence: A Walkthrough from Smart Things to Smart Hyperspaces and UbiKids", *Int'l Journal of Pervasive Comp. & Comm.*, 1(1), Mar. 2005.

# Kallima: A Tag-Reader Protocol for Privacy Enhanced RFID System

Yusuke Doi, Shirou Wakayama, Masahiro Ishiyama, Satoshi Ozaki, and Atsushi Inoue

Communication Platform Laboratory,  
Corporate Research & Development Center, TOSHIBA Corporation,  
1 Komukai-Toshiba-cho, Saiwai-ku, Kawasaki, Kanagawa, Japan  
{ydoi, shirou, masahiro, fe, inoue}@isl.rdc.toshiba.co.jp

**Abstract.** Privacy is a major concern with RFID tags and many solutions have been proposed. As many approach requires secure hash function on each tag, cost of tags imposed by those solutions is significantly high for wide development. We propose a protocol that uses pre-calculated Bloom filter to send tag identity for increased privacy with little additional cost per tag. In our approach, secure hash function calculation is done in tag production phase and each tag does not have any hash functions. Instead, each tag must have random number generator and volatile memory.

## 1 Background

RFID (Radio-Frequency IDentification) tags are emerging devices that make automatic identification more convenient. However, convenient automatic identification is sometimes dangerous with respect to privacy. Some consumer groups oppose to RFID due to the privacy infringement. If a store uses RFID tags for inventory control and leaves the tags on the items after the checkout, customers themselves may be traced by the RFID tags. Tags on some personal belongings like purses, shoes, and clothes can help tracking a customer without being noticed. In this paper, we call this problem *owner tracing*. To keep user's privacy, one can use high-end tags. Some of high-end tags like Felica<sup>1</sup> may provide high level security using cryptographic function.

On the other hand, low cost tags are needed for wide deployment of RFID systems. Because a major objective of RFID is cost reduction of item handling in distribution systems, RFID system must be low cost as much as possible to deploy. Peoples in Auto-ID Labs protests that a tag must be as cheap as 5 or 10 cents for wide deployment[1]. At the same time, a 5 cents tag is a difficult target to accomplish[2]. Hence, RFID tag system that aims wide deployment can afford very little cost for security per tag.

### 1.1 Assumptions

We assume the following entities in our RFID system model.

A victim has an item. The item has an RFID tag attached on it. RFID tag sends signals on read request.

---

<sup>1</sup> <http://www.sony.net/Products/felica/>

An attacker has an RFID reader that can read signal of RFID tags within its range. We assume the attacker has no other clue than RFID signal to identify what item is in the range. For example, attacker may use a video camera with the reader to bind RFID signals and people. In case if an attacker can bind a RFID signal with the victim, the attack becomes easier. The case is discussed in the section 4.1.

We also assume the victim and other people appear within the attacker's range in random pattern. The reader captures many signals from those people and the attacker has no knowledge which one is from the victim's. In addition, a reader captures other casual signals from other tags that comes by chance. We assume arrival pattern and signals from those tags are random.

Owner tracing occurs as the reader successfully identify the signal from the victim's item. More precisely, attacker receives a sequence of signals from the reader. The attacker can trace a victim if the attacker can find relation between two or more signals of the victim's tag out of the sequence. In other words, owner tracing fails if attacker cannot find relation between the sequence of signals from the reader.

In addition, our model of trust assumes that a reader is a legitimate reader for a RFID tag if the reader knows the tag's identifier. The identifier itself is the shared secret to make trust relationship between the tag and the reader.

## 1.2 Our Goals

Our goal is to enable the development of RFID tags that has 1) resistance against owner tracing 2) with smallest additional cost per tag.

## 2 Related Works

Many studies on the prevention of owner tracing have been carried out. Some solutions reduce usability while others increase cost per tag by complex hardware for secure hash function.

Weis et al.[3] and Juels et al.[4] addressed the security problems of RFID systems and proposed some solutions for low-cost tags. One of their solutions (hash-lock) utilizes the one-way hash function. Due to the difficulty of implementing the hash function in small hardware logic (e.g. the SHA-1 hash function requires 20K to 50K gates), this solution entails a small but significant extra cost. In addition, this approach has a weakness in the unlocked state, that is, attackers can override the tag's authority in the unlocked state. Attackers may also turn their attention to the readers, because the readers may leak the secret if an attacker repeats a metaID sent by a locked tag. Although these attacks are detectable, it is also possible for denial-of-service attacks to be attempted against whole systems. Another solution (blocker tag) blocks the anti collision process to keep the tag ID be anonymous. However, the blocker tag itself has no intelligence and works indiscriminately. This may cause trouble as some blocker tag may block authorized readers.

A proposal from Ohkubo et al.[5] uses randomized hash chain to protect tag privacy. At  $i$ th read, a tag sends  $a_i = G(s_i)$ . At the same time,  $s_{i+1} = H(s_i)$ . Both  $H$  and  $G$  are different secure hash functions, and  $s_i$  is secret of the tag. Their argument describes forward security against tracing. Even if an attacker seized secret  $s_i$  of the tag,



the attacker cannot find past traces out of read records. With their proposal, a tag system can provide forward security. However, their approach also requires hash function implemented inside the tags.

### 3 Our Approach

As a step toward cheap RFID tags with privacy, we utilize a pre-calculated Bloom filter[6] in the protocol between RFID tags and readers. In this section we propose a tag-reader protocol called Kallima.

To prevent unauthorized readers from tracing a tag, communication between tags and readers must have no fixed id in plain form. In Kallima, the tag sends a Bloom filter with noise (NBF: noisy Bloom filter) and the reader tests its known IDs against the filter. We describe the protocol in the following order: bloom filter, components, tag identification, and anti collision.

#### 3.1 Bloom Filter

A Bloom filter is a simple data structure that utilizes a set of secure hash functions and represents a data set  $S$ . We adopt it because it is resistant to one-way noise to make bits of value 0 to 1. With the resistance, a tag can disguise its identity in noise.

Generally speaking, one can test a Bloom filter of a set  $S$  if a data  $x$  is in  $S$ . The test yields a positive or negative result. A negative result means that  $x$  is definitely not in  $S$ . However, there is a risk of false-positives, which has a tradeoff relationship with space efficiency.

In Kallima, a Bloom filter is created from a tag ID  $x$  alone. A Bloom filter that represents  $S = \{x\}$  is called SBF. This Bloom filter is called a seeding Bloom filter (SBF). A SBF is created in the following steps.

First, a bit array  $B$  of length  $m$  is initialized to 0. Then, a set of  $k$  independent secure hash functions  $h_1, \dots, h_k$  with range  $\{1, \dots, m\}$  is applied to  $x$ . The hash value of  $x$  is assumed to be distributed uniformly between 1 and  $m$ . Finally, for each hash value  $v$ , the corresponding value of the bit array is set to 1. The result  $B$  is a SBF for  $x$ .

Receivers of a Bloom filter can find source tag ID of the filter if they know a tag ID  $y$  by testing  $y$  against the filter. Testing on a Bloom filter is performed in similar steps with Bloom filter creation steps. The same set of hash functions used in creation steps is applied to  $y$ . For each hash value  $v$ , the corresponding bit of the filter is checked. The test result is positive if all the checked bits are 1. The result is negative otherwise. In other words, tests with  $B \& (1 \ll v) == 1$  for all  $v = h_i(y)$  in  $i = \{1 \dots k\}$  yields a positive result.

A false positive occurs if all the hashed values of  $y$  are 1 by chance. In our case, noise can make some random bits to have value of 1. Figure 1 shows an example (each cube represents a bit and shadowed cubes are set to 1). In this case,  $h_1(x) = h_2(y)$ ,  $h_2(x) = h_1(y)$ , and  $x \neq y$ . As a result,  $y$  seems to be in  $S$ . Actually it is not. The risk of a false positive for a Bloom filter is given as  $s^k$ , as  $s$  (saturation) is ratio of bits set to one among all bits in the filter. For example, saturation of filter 0b01101010 is 0.5. If  $k = 3$ , false positive rate is 0.125.

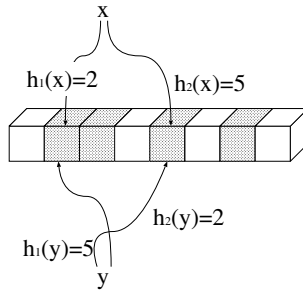


Fig. 1. An example of a false positive on a Bloom filter 0b01101010

### 3.2 Tags and Readers

Figure 2 shows an example of abstract block diagram of a tag. A tag’s memory has two types of identifiers written in it. The tag ID and static Bloom filters calculated from the ID (SBFs: Source Bloom Filters). A SBF is calculated by standard computer at the time of tag production.

Note that a tag may have one or more SBFs with different parameters. We assume each SBF belongs to a *class* that defines the parameters for calculating the SBF. For example, a class A SBF may have  $m = 512, k = 7$  and  $H_n(d) = SHA1(d + n)$ ; a class B SBF may have  $m = 1024, k = 13$ , and the same  $H_n(d)$  as in class A; and so on.

The simplest tag consists of a controller, a read-only memory for the ID, an RF module, and a one-way noise imposer. The one-way noise imposer is a method to disguise tag’s identity. Other modules are the same as those in regular RFID tags without any security mechanism.

The one-way noise imposer has a simple function. It imposes one-way noise on the input bit streams. The one-way noise imposer turns 0 at a random place in the input bit stream into 1 until the bit stream saturation  $s$  comes to predefined value. We assume an expected saturation of the output bit stream as  $s = 0.5$  for example.

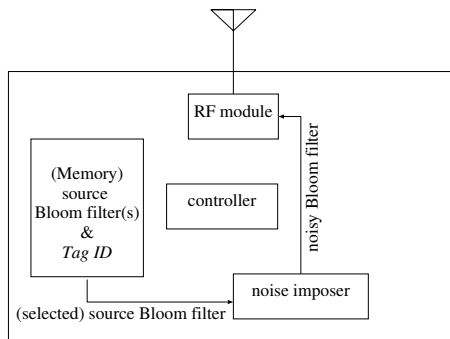
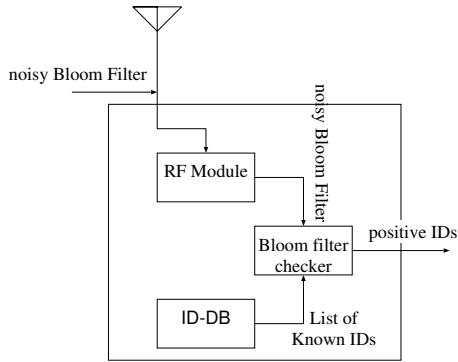


Fig. 2. Block diagram of a tag with Kallima



**Fig. 3.** Block diagram of a reader for Kallima

To turn an SBF of the tag into a bit stream looks random, the SBF is sent to the one-way noise imposer. The output is called a noisy Bloom filter (NBF). A Bloom filter does not become broken even if any additional bits are set to 1. A side effect of the added noise is an increased false positive rate ( $(\frac{k}{m})^k$  becomes  $s^k$ ).

Figure 3 shows an example of abstracted structure of a reader. A reader consists of an RF module, an ID-DB, and a Bloom filter checker. The ID-DB and the Bloom filter checker are the modules specially required for Kallima; the other modules are the same as those in ordinary RFID tag readers.

The ID-DB contains a list of tag IDs. In our model, a reader is an authorized reader for a tag if the tag's ID is in the reader's ID-DB. Otherwise, the reader is not authorized to identify the tag.

### 3.3 Identification of a Tag

Readers receive NBFs. They must test the filter to find tag IDs in their communication range.

The reader uses the Bloom filter checker to find candidate tag IDs. In the simplest implementation, it tries the known ID in the ID-DB one by one against the received NBF. The filter check process returns a set of candidate IDs or an empty set.

The process can produce three results that can lead for four implementation patterns.

- The process results in an empty set: there are no known tags
- Only one ID in the candidate set: the candidate ID as the detected tag (may perform validation process)
- Two or more IDs in the candidate set:
  - report entire candidate IDs
  - perform conflict resolution in some way

The conflict resolution process and validation process is not described in this paper.

The saturation of NBFs is controlled by one-way noise imposer, and the reader can estimate how false positives may occur. For example, if a reader knows 100 tag IDs, a 0.01% false positive rate corresponds to one expected read error among 100 trials.

If this risk is not negligible, the reader may request tags to send NBFs with increased  $m$  and  $k$  (i.e. NBFs of a higher class). If a reader knows  $10^6$  IDs and its application accepts only 1 misdetection for  $10^6$  reads, the situation requires an FPR of as low as  $10^{-12}$ . This can be achieved with a Bloom filter with  $k = 40$  if the filter saturation is 0.5 because  $FPR = s^k = 0.5^{40} < 10^{-12}$ .

An additional note is that the filter length  $m$  has a lower impact than in regular uses of Bloom filters as long as  $k$  is far lower than  $m$ . This is because the false positive rate is strongly bound with  $s$  and  $k$ , and the  $s$  of an NBF can be controlled by the one-way noise imposer. Discussions on  $m$  against  $k$  is described in section 4.1.

### 3.4 Anti Collision and Temporary ID

If the above protocol does not transmit any ID at all, application of ALOHA-style anti collision[7] is difficult. To avoid the problem, a tag should have a temporary ID that is valid for a period. Because tags have random number generators, it is easy to generate a randomized temporary ID at the beginning of communication. The temporary ID is held in a temporary memory such as an SRAM with capacitor, and is reused until the memory expires. Usually, ALOHA-style anti collision provide continuous energy until the process finishes.

## 4 Discussions

Here we discuss how our proposal achieve the goals outlined in section 1.2. We discuss the first goal in subsection 4.1 and the second goal in subsection 4.2 below.

### 4.1 Strength Against Owner Tracing

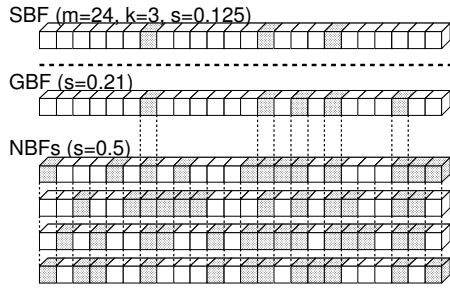
Compared to hash-lock and randomized hash chain, our approach has limited strength against owner tracing. In this section we describe when an attacker can trace tag and its owner.

An unauthorized reader for a tag is that do not know the tag's ID. In our model, the tag only sends NBFs and owner tracing occurs as an attacker reveals relation between one NBF and another.

If an attacker can take multiple NBFs from a tag, identification of an NBF becomes easy. First, an attacker collects two or more NBFs from the victim's tag. Then the attacker takes bitwise-and between all the NBFs. The bit stream produced by the operation contains less noise. The produced filter is called a guessed Bloom filter (GBF).

A well-produced GBF contains all the bits set to 1 in the SBF and some noise. Because  $k$  of the filter is a known number, the attacker can guess how much noise is left in the GBF and the certainty of each bits set to 1 in the GBF. Thus, with a GBF, the attacker can estimate if another anonymous NBF is related to the GBF. Figure 4 shows an example GBF constructed using four NBFs.

To make this kind of attack difficult, a tag should avoid to give multiple NBFs to attackers. In our assumptions, readers have no knowledge in advance to find a couple of NBFs out of many has come from a tag. If NBFs from a different tag are mixed in, half (actually,  $s$ ) of the traces of the SBF of the victim's tag will be lost in the produced



**Fig. 4.** An example of GBF

GBF. Hence, as long as the reader cannot isolate the victim’s tag from the other tags, received series of signals are not usable to make a GBF.

To prevent attackers from obtaining multiple NBFs easily, a previously used NBF may be kept in a static memory with a reasonable lifetime. Because a tag needs a static memory to store the temporary ID for anti collision (described in subsection 3.4), the previously used NBF may be stored instead of temporary ID to make anti collision process.

Too long lifetimes, such as a day, help attackers to trace tags using the NBF or temporary ID, and too short lifetimes result in an increased chance for attackers to have successive access to an isolated tag. Expected period of time of tag isolation is longer if tags are placed or carried alone. If application expects many tags in an area the period of time can be short.

In addition to correlation of two NBFs, there are two ways that enables ID leakage. The first is leakage from an authorized tag reader, while the other is ID calculation from an anonymous NBF. Leakage from readers is beyond the scope of this paper because it includes various topics ranging from tamper-resistant hardware for readers to laws against intentional secret leakage.

The other mechanism is brute force attack. Our approach is not strong enough if an attacker deploys more computing power to track an ID.

An attacker with an NBF may try a brute force attack to find out the tag’s ID. However, brute force does not help much in the situation because of the false-positives and the vast amount of tag ID data. For an NBF with  $k = 40$  and  $s = 0.5$ , even a 64-bit long ID space can result in more than 16 million ( $0.5^{40} \times 2^{64}$ ) IDs as candidate IDs.

It is clear that 16 million candidate IDs out of 64-bit long ID space is enough to identify a tag out of limited size of NBFs. If a newly received NBF is from the victim’s tag, at least one of candidate IDs matches. If attacker receives an anonymous NBF, it may match with one or more candidates with as much possibility as  $(1 - (1 - s^k)^{n_c})$  where  $n_c$  is number of candidate IDs. If  $k = 40$ ,  $s = 0.5$ , and ID width is 64bits, only one NBF out of more than 65000 NBFs returns one or more candidate IDs as positive.

The attacker may use two or more NBFs from the tag to find small set of ID candidates. In that case, the risk of owner tracing will be described as follows.

Each NBF has possibility of false positive rate of  $s^k$ . With  $i$  NBFs from the same tag, false positive rate becomes  $s^{m \cdot i}$ . For two NBF with  $k = 40$  and  $s = 0.5$ , one can find a 64-bit ID string that matches both NBFs with high probability ( $0.5^{80} \times 2^{64} \ll 1$ ).

## 4.2 Tag Complexity

We believe tags for Kallima is simple enough for low cost manufacturing because our approach works without secure hash function and persistent memory on tags. As the result from previous discussions, a tag with Kallima should have a static memory as large as a maximum class of NBF, a capacitor to maintain the memory, and a one-way noise imposer in addition to the regular tag components.

For comparison, hash-lock requires secure hash function in addition to rewritable persistent memory to keep metaIDs. Randomized hash chain also requires hash function and rewritable persistent memory to keep ID secret.

## 5 Conclusions

In this paper, we discussed a solution for privacy issues on RFID systems of 5-cent tags. We can solve privacy issues on RFID systems as long as we can afford high price tag that uses complex hash function. However, we cannot afford it on low end system for now.

The challenge is how to hide tag identity without hash, and we propose combination of one-way noise with Bloom filter. As a solution, we propose Kallima, a tag-reader protocol that utilizes Bloom filter with noise as tag identifier. Because tags with Kallima does not need any secure hash functions and persistent rewritable memory on each tag, we expect Kallima can be as cheap as simplest tags used in the market.

## References

1. Sarma, S., Brock, D.L., Ashton, K.: The networked physical world. Technical Report MIT-AUTOID-WH-001, MIT Auto-ID Center (2000)
2. Sarma, S.: Towards the 5c tag. Technical Report MIT-AUTOID-WH-006, MIT Auto-ID Center (2001)
3. Weis, S.A., Sarma, S.E., Rivest, R.L., Engels, D.W.: Security and privacy aspects of low-cost radio frequency identification systems. In: Lecture Notes in Computer Science. Volume 2802. (2003)
4. Juels, A., Rivest, R.L., Szydlo, M.: The blocker tag: Selective blocking of rfid tags for consumer privacy. In: Proceedings of the 8th ACM Conference on Computer and Communications Security, ACM Press (2003) 103–111
5. Ohkubo, M., Suzuki, K., Kinoshita, S.: Cryptographic approach to a privacy friendly tag. In: RFID Privacy Workshop, MIT (2003)
6. Broder, A., Mitzenmacher, M.: Network applications of bloom filters: A survey. In: Proceedings of the 40th Annual Allerton Conference on Communication, Control, and Computing. (2002)
7. Finkenzeller, K.: RFID-Handbook, 2nd edition. Wiley & Sons, Ltd (2003)

# Selective Collision Based Medium Access Control Protocol for Proactive Protection of Privacy for RFID

JuSung Park, Jeonil Kang, and DaeHun Nyang

Information Security Research Laboratory,  
INHA University, Korea\*  
{security, dreamx}@seclab.inha.ac.kr, nyang@inha.ac.kr  
<http://seclab.inha.ac.kr>

**Abstract.** RFID is rapidly being deployed because of its versatility. However, the privacy problem cannot be handled effectively because of limited capability of RFID tags. We propose a secure medium access control(MAC) protocol to solve the privacy problem. Our secure MAC protocol defines a special kind of tag called an “ownership tag”. The singulation procedure is well defined only if the ownership tag is involved in the singulation process. Instead of scrambling the information of ordinary tag using the ownership tag’s key information and just sending ownership tag’s information in clear form, we use forced collision generated by the ownership tag to let a reader know garbage bits that are inserted in ordinary tags. Security and performance analysis for the protocol are provided.

## 1 Introduction

RFID(Radio Frequency IDentification) is a very useful tool for speeding up supply chain management. RFID tag can be attached to every product if the product is required to be identified in a computerized way and it will make the bar-code system obsolete sooner or later.

The advantage of RFID is not only identification without contact, but also huge amount of information that the tag can hold. The size of memory RFID tag is much larger than that of bar-code and thus, RFID tag can identify every instance of products as well as every kind of products. These advantages of RFID works like double-edged sword, and they might be very harmful if abused. Thus, privacy problem such as stolen information of person and industry espionage have been raised continuously. The privacy problem cannot be handled effectively because of limited capability of RFID tags: the size of memory of an RFID tag is too small to store some useful information such as long keys and also a tag has very small number of gates to perform cryptographic operation such as one-way hash function, exponentiation, etc. Beside that, because passive tag does not have power supply in itself, external device such as a tag reader must supply power for proper operation of tags. All the limitations make the privacy problem hard to be solved.

---

\* This work was supported by the Korea Research Grant funded by Korean Government(R03-2004-000-10023-0).

In this paper, we propose a secure medium access control(MAC) protocol to solve the privacy problem. Our secure MAC protocol defines a special kind of tag called an “ownership tag”. Our protocol is defined assuming the binary tree walking, but it can be extended to ALOHA-based MAC protocol. In section 2, we propose and discuss our secure MAC protocol in detail. In section 3, we consider the performance of the RFID system that adopts our scheme. In section 4, we analysis estimated security strength of our scheme. Finally, in section5, we summarize our whole paper, and discuss some possibilities of our scheme.

## 2 Ownership Tag

### 2.1 Overview

The goal of our scheme is to prevent the leakage of tag’s information at any time except when the consumers want to offer tag’s information, which is a kind of “proactive privacy protection”. To do that, a tag’s information should be encoded by some cryptographic operation, and the decoding key has to be provided. In our secure MAC protocol, the key is provided as a special kind of tag called an “ownership tag”, and the ordinary tags are protected by inserting some bogus bits that are not known except when the ownership tag is not located nearby the ordinary tags. Due to these features, the ownership tag must be present in nearby, when a reader wants to read some tag’s information correctly. Instead of scrambling the information of ordinary tags using the ownership tag’s key information and just sending ownership tag’s information in clear form, we use forced collision generated by the ownership tag to let a reader know garbage bits that are inserted in ordinary tags. This approach has more security against passive eavesdropper than that because an attacker must eavesdrop a whole session to recover the key information. If key information is sent in clear form by a special tag, an eavesdropper can obtain easily the key information by listening only the special tag’s communication. On the contrary, key information of our secure MAC protocol is dispersed through the whole session. The ownership tag is a kind of RFID tag that has two antennas for the forced collision like the blocker tag, and thus, can operate as a blocker tag if proper logics are implemented. Note that forced collision in our proposal works for revealing the key information, whereas forced collision in the blocker tag is used for hiding tag’s ID.

Ownership tag wrapped by a foil-receipt can be provided to customers. When a customer wants to make ownership tag enable, he can unfold the receipt to prevent RFID from being read.

### 2.2 System Requirements

Our protocol assumes the followings:

- Ordinary tag’s memory consists of OTP (One Time Programmable) memory and ROM (Read Only Memory). Directive to decide whether a bogus bit is to be inserted or not are saved in OTP, while tag’s data are saved in ROM.



**Table 1.** Component of ownership tag scheme

Part	Components	Role
Reader	Reader	reads tag's information using binary tree walking algorithm
	Antenna	listens a query from a reader and send response
Ordinary Tag	ROM	stores tag's <i>data</i>
	OTP	stores directive <i>input_directive</i> to decide whether a bogus bit is to be inserted or not
	3 Pointers	<i>bogus_pointer</i> points to position value of bogus bits. <i>data_pointer</i> points at tag's data. <i>input_bogus_pointer</i> points at inserted bogus data.
Ownership Tag	2 Antennas	make a forced collision
	OTP	stores the same directive as that of ordinary tag
	Pointer	operates in the same way as the <i>input_bogus_pointer</i> of ordinary tag

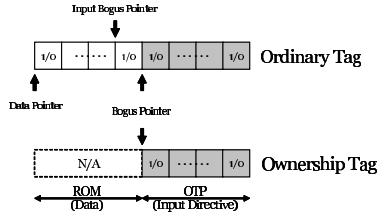
- RFID tag is passive type. That is, it does not have any power supply in itself.
- Reader and tag use a binary tree walking for singulation.
- Ownership tag has OTP memory that is cheaper than EEPROM or flash memory.
- Ownership tag has two antennas in order to make a forced collision.
- Our scheme can adopt the silent tree walking for protection against trivial eavesdropping[3].

Forced collision is used both by blocker tag and ownership tag, but the purpose is quite different. Forced collision made by ownership tag works for revealing the key information, whereas forced collision in the blocker tag is used for hiding tag's ID.

### 2.3 Initialization

The ownership tag and the ordinary tags must be initialized as **figure 1** which is described as follows.

- After reader obtains ordinary tag's *data* using the binary tree walking scheme, it inserts randomly generated *input\_directive* into the OTP of ordinary tag, where the *input\_directive* decides whether a bogus bit is to be inserted between ordinary tag's information bits or not.
- Ordinary tags have 3 pointers to be initialized. One is called *data\_pointer* that is initialized to point to the first bit of tag's data in ROM. The second one is called *input\_bogus\_pointer* that points to bogus bits to be inserted, which is initialized to point to the end of data bits and then decreased one by one whenever a bogus bit is inserted. The last one is called *bogus\_pointer* that is initialized to point to the first bit of *input\_directive* in OTP.
- An ownership tag has 1 pointer and *input\_directive* like an ordinary tag, which are initialized in the same way as that of ordinary tags.



**Fig. 1.** Initialization of the ordinary tag and the ownership tag. The ordinary tag has *data* and *input\_directive* with three pointers, and the ownership tag has only *input\_directive* with one pointer.

### 2.4 The Protocol

We describe our protocol by showing each entity’s role.

**Ordinary Tag.** Ordinary tag has its own data in ROM and *input\_directive* in OTP memory, so its answer for the query looks scrambled if the ownership tag does not involve in the singulation process. To achieve this, we define operation of ordinary tag as following:

- If the number of reader’s query is odd, ordinary tag responds with its data bit that *data\_pointer* points to in ROM. Or else, it refers to *input\_directive* in OTP memory.
- If the bit of *input\_directive* is ‘0’, ordinary tag regards it as ‘null’ and then, it responds with its data bit that *data\_pointer* points to.
- Else if the bit value is ‘1’, tag regards it as ‘input bogus command’ and then, it responds with bogus bit that *input\_bogus\_pointer* points to.

Note that the bogus bits are not randomly generated bit sequence but reverse sequence of tag’s data bits. According to the ordinary tag’s behavior, even though tags have the same *input\_directive*, they will produce different bogus bits respectively. Pseudo-code for operation of ordinary tag is followed:

```

loop
  QUERY := receive()
  if QUERY == odd-th //Data transmission
    if *data_pointer == EOF
      break loop
    else
      respond(*data_pointer);
      data_pointer++;
  else //Bogus bit transmission
    if input_directive == 0
      bogus_pointer++;
      respond(*data_pointer);
    
```

```

data_pointer++;
else
    bogus_pointer++;
    respond(*input_bogus_pointer);
    input_bogus_pointer--;

```

**Ownership Tag.** Ownership tag is in one of three states: state to make a forced collision, state to listen reader’s query, and state to keep silence. It makes a transition to one of three states according to reader’s query. Following pseudo-code shows how ownership tag works.

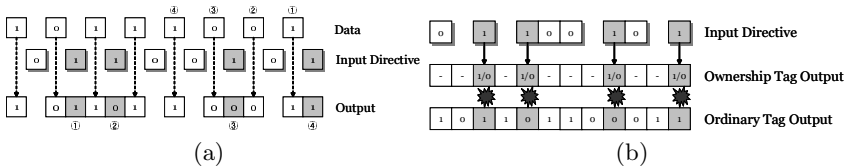
```

loop
    initialize bogus_pointer
    loop until one singulation completes
        QUERY := listen() /* from reader */
        if QUERY != re query && QUERY == even-th
            if *input_directive == 1
                send(collision)
            bogus_pointer++

```

Because an ordinary tag responds with a data bit that *data\_pointer* points to for the odd numbered query, ownership tag does not respond for the ordinary tag to send its data to a reader. For the even numbered query, ownership tag makes collision according to the *input\_directive* selectively. Also, if the query is received again after the collision, it must keep silent for the reader to know the existence of bogus bit.

**Reader.** Reader recognizes the bogus bit position of all involving tags during collision resolution process by sending re-query and checking whether the response comes or not. If there is at least one ‘no response’ for all re-queries in the same level where collision has occurred, reader can find that the level with collisions contains a bogus bit. However, even though ownership tag keeps silence, if one of ordinary tags responds with some data, reader will regard this position as a data bit. Actually, the position is for bogus bit, so we must be careful when a reader inserts *input\_directive* into ordinary tags.



**Fig. 2.** (a). How an ordinary tag works in our scheme. The ordinary tag’s *data* are scrambled by its *input\_directive*. (b). An ownership tag helps the reader be able to read the ordinary tags’ *data* correctly.

The *input\_directive* must satisfy some property to avoid the above malfunction. When reader generates *input\_directive*, reader should avoid inserting bogus bit into the level  $h$  of the binary tree that has  $2^h$  nodes (or  $2^h$  collisions) fully. **Figure 3** shows binary tree constructed from the binary tree walking algorithm. Because the number of tags that involve in singulation process usually much less than the total number of possible data, we do not need to perform collision resolution procedure after resolving  $h_1$  bits. Thus, the tree looks like **figure 3** when the probability distribution of data is uniform.  $h_1$  is approximately  $\lceil \log_2 n \rceil$  ( $n$  = number of tags). Assuming that  $n$  is 128,  $h_1 \approx 7$  bits. Namely, a reader must avoid inserting bogus bits into the first 7 bits of tag.

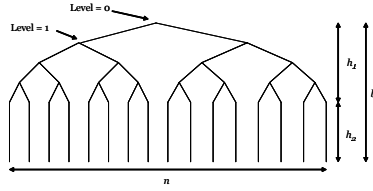
Bit positions that bogus bits must not be inserted into are easily decided by a reader during its initialization phase. After reading original data of all tags during initialization, a reader can find bit positions where collisions occur. Except the level  $h$  of the binary tree that has  $2^h$  nodes fully, bogus bits can be freely inserted into any level that has less than  $2^h$  nodes. If we insert a bogus bit into the level  $h$  that has less than  $2^h$  nodes, ownership tag can keep silence without being disturbed by other ordinary tags in any empty branch. Thus, the  $i$ -th bit of *input\_directive* must be '0' if level  $i$  has  $2^i$  nodes. Otherwise, reader fills the remaining bit positions randomly.

Reader operates as follows.

```
// begin binary tree walking
// collision_position_list : a stack for depth first search

input_directive[] := 0
send(next-bit query)
loop
  BIT := read()
  if BIT == EOF
    save(tag's data)
    if collision_position_list != empty
      pop(collision_position_list)
      send(wake-up query)
    else
      break loop
  else if BIT == collision
    push(collision_position_list, current bit position)
    send(re query)
  else
    send(next-bit query)
    input_directive[current depth] := 1

// begin process of inserting bogus bits
while i < bit length of input_directive
  if input_directive[i] == 1
    input_directive[i] := randomly generated bit;
```



**Fig. 3.** Binary Tree Walking Scheme working Model. After  $h_1$ , the reader communicate with one tag by one-bit.

### 3 Performance Evaluation

In accordance with **table 2** and **figure 3**,  $h_1 \approx \log_2 n$  and  $h_2 \approx b - \log_2 n$ . Thus, amount of bit transferred in  $h_2$  are

$$2n(b - \log_2 n) \tag{1}$$

Amount of bits transferred in  $h_1$  by ‘wake-up’ query is

$$m \sum_{k=1}^{\log_2 n - 1} = m(n - 2) \tag{2}$$

In  $h_1$ , sum of ‘next-bit’ query and responses is

$$2 \sum_{k=1}^{\log_2 n} = n - 4 \tag{3}$$

Thus, total sum of bits transferred in binary tree is

$$(2b - 2\log_2 n + m + 1)n - 2m - 4 \tag{4}$$

We can say that it take  $(2b - 2\log_2 n + m + 1)n - 2m - 4$  BT to singulate  $n$  number of tags. Because total sum of IBTs required is the same as that of (4) if we let  $m = 1$ , the sum of IBTs in binary tree is

$$2(b - \log_2 n + 1)n - 6 \tag{5}$$

**Table 2.** Component of ownership tag scheme

Notation	Explanation
$b$	The number of bits of tag
$m$	Bit length of $b$ , or $\lceil \log_2 b \rceil$
$n$	The number of tags
BT(bit time)	Unit time required for transmitting a bit
IBT(inter bit time)	Unit time between BTs

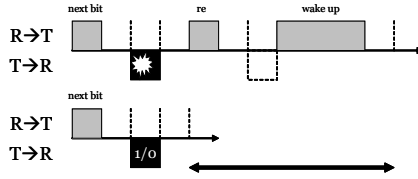


Fig. 4. When a collision occurs, it requires more BT and IBT to singulate

Table 3. Comparison of singulation time,  $n = 100$

Total Length (+bogus bits)	Un-scrambled Ordinary tag		Protected tags with- out ownership tag		Protected tags with ownership tag	
	BT	IBT	BT	IBT	BT	IBT
64(+32)	12157	11667	18655	18067	44255	28255
128(+64)	25055	24467	37953	37267	95553	57153
256(+128)	50573	50067	76451	75667	204451	114851
512(+256)	102051	101267	153349	152467	434949	230149

As shown in figure 4, when ownership tag is involved in singulation, amount of time required to singulate tags increase by  $m + 1$  BTs and 3 IBTs for one forced collision. Remind that reader should not insert bogus bits into the  $h_1$ . If we let the number of inserted bogus bits  $c$ , which must be inserted bogus bits into  $h_2$  only, then the number of added collision is in proportion to  $c$  and  $n$ . If we denote amount of time required to send all bits by  $BT_{off}$  and  $IBT_{off}$  when ownership tag is absent, or else  $BT_{on}$  and  $IBT_{on}$ , then

$$BT_{on} = BT_{off} + cn(m + 1) \tag{6}$$

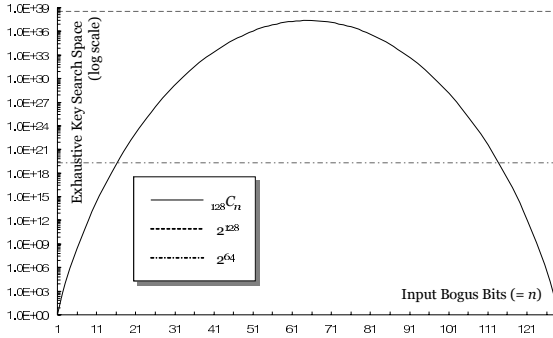
$$IBT_{on} = IBT_{off} + 3cn \tag{7}$$

$BT_{off}$  and  $IBT_{off}$  are the same as (4) and (5) respectively.

Ownership tag increases BT and IBT by almost 50% ~ 250%, but the degradation is not so significant. After reader detects a bit position of a bogus bit, it can ignore the same bit position of other tags, because all the tags have bogus bits in the same bit positions. In such a case,  $n = 1$  or 2 in (4). Using the idea, only the added overhead is not 5% ~ 250%, but the same as addition of one or two scrambled ordinary tags.

## 4 Security Analysis

Without ownership tag, neither any reader nor attacker can extract tag’s information because the information is scrambled with randomly generated *input\_directive*. Even though every tag has the same *input\_directive* with each other, inserted bogus bits are different. Thus, attacker who has collected many tag’s information scrambled with the same *input\_directive*, it is hard to find out



**Fig. 5.** Key Space for exhaustive search according to the number of inserted bogus bits

exact position of bogus bits because any same bit pattern does not appear in the scrambled data. Actually, this scrambling is a kind of simple block cipher using substitution with expansion and transposition. If higher level of security is required at the expense of cost-effectiveness, other strong block ciphers can be used instead of this scrambling. In that case, reader can recover (or decrypt) all tags’ information only after it finds all bits of *input\_directive*(or a key).

**Figure 5** shows the size of exhaustive key search space for the number of inserted bogus bits. The size of key search space against ownership tag is  $\text{MIN}\{{}_{128}C_n, 2^n\}$  if we use 128-bit *input\_directive*, and  ${}_{128}C_n > 2^n$  for  $n > 0$ , where  $C$  denotes combination operator. Thus, the size of key search space of brute force attack is  ${}_{128}C_n$ . As shown in Figure 5, we can obtain the highest level of security when the number of bogus bits inserted is 64 bits. The number of actually inserted bogus bits will be 64 bits on average, if we use random number generator for *input\_directive* that has uniform probability distribution.

If reader detects exact position of a bogus bit during singulation process owing to ownership tag, it does not need to send any re-query for the bit position because he already knows that other tag’s response for query of the bit position is bogus. This property makes our scheme more secure and more efficient. That is, if illegal reader (eavesdropper) wants to know all positions of bogus bit, it must eavesdrop whole communication session between tags and a legal reader. In that sense, our scheme can be seen as a MAC protocol that defines how to send key information secretly through open channel.

## 5 Conclusion

We propose a secure MAC protocol, which defines a special tag called ownership tag. Only one who holds the ownership tag can insist his ownership of products with RFID tags paired with the ownership tag. From a customer’s point of view, ownership tag is more proactive than the blocker tag to protect privacy, because information of tags is scrambled at normal times and is in clear form only when ownership tag is involved in the protocol. Only when customer needs insist or

identify his tags, he will carry the ownership tag. If customer wants to hide ordinary tags while carrying the ownership tag, the ownership tag should be in a faraday cage, which can be provided as a form of receipt.

Actually, the concept of ownership tag can be easily implemented by making ownership tag send key information in clear form using normal binary tree walking. However, our scheme has more security against passive eavesdropper than this because an attacker against our scheme must eavesdrop a whole session to recover the key information. If key information is sent in clear form by a special tag, an eavesdropper can obtain easily the key information by listening only the special tag's communication. On the contrary, key information of our secure MAC protocol is dispersed through the whole session.

Though we show only binary tree walking version of ownership tag, it can be adopted to the ALOHA protocol.

## References

1. Ari Juels, Ronald L. Rivest and Michael Szydlo : The Blocker tag: Selective Blocking of RFID Tag for Consumer Privacy, RSA Laboratory, MIT
2. Ari Juels and John Brainard : Soft Blocking: Flexible Blocker Tags on the Cheap, RSA Laboratory
3. Stephen A. Weis, Sanjay E. Sarma, Ronald L. Rivest and Daniel W. Engels : Security and Privacy Aspect of Low-Cost Radio Frequency Identification Systems, In Security in Pervasive Computing, 2003
4. Stephen A. Weis : Security and Privacy in Radio-Frequency Identification Devices, MIT, 2003



# *iCane* – A Partner for the Visually Impaired

Tsung-Hsiang Chang, Chien-Ju Ho, David Chawei Hsu, Yuan-Hsiang Lee,  
Min-Shieh Tsai, Mu-Chun Wang, and Jane Hsu

National Taiwan University, Computer Science and Information Engineering  
<http://www.csie.ntu.edu.tw>

National Taiwan University, Electrical Engineering  
b90106@csie.ntu.edu.tw, b90090@csie.ntu.edu.tw,  
bengiu.david@gmail.com, b90097@csie.ntu.edu.tw,  
b90070@csie.ntu.edu.tw, b90082@csie.ntu.edu.tw,  
yjhsu@csie.ntu.edu.tw  
<http://www.ee.ntu.edu.tw>

**Abstract.** Any modern society should be concerned over the well-being of the visually impaired citizen. It is the responsibility of the society to lessen the inconvenience and anxiety experienced by the handicapped. In addition to helping one to avoid obstacles on the sidewalks, technology can further help in navigating to places. In this paper we attempt to create a supportive environment with timely and useful information to guide the visually impaired to comfortably roam in a city that cares.

## 1 Introduction

To navigate on the streets today, the visually impaired is required to walk with the aid of *white canes* and *tactile paving*. The standard white canes are used for detecting obstacles, and tactile paving indicates the general configuration (e.g. straight path, intersection etc.) of the street in front. However, it does not provide detailed information about the pedestrian's current *location*. The main idea of the proposed design is to enrich the interaction of the visually impaired with the environment using smart sensors, intelligent software and accessible interface. The *iCane* is designed to provide its user location-based services, e.g. navigational guidance, points-of-interest referral, and mobile commerce, through a friendly voice interface.

*iCane* equips the standard white cane with an RFID (radio frequency identification) reader 0, which connects to a personal digital assistant with Bluetooth headphones. The reader retrieves location and relevant information, e.g. intersection, elevator and stairs, nearby shops etc., from RFID tags embedded in the environment. The user would give speech command to indicate the desired destination. The navigation service then computes the optimal path to the destination based on the current location and a map database. Navigational advice and other pertinent information are delivered to the user through the headphones at appropriate times.

The features of *iCane* are summarized as follows.

- *iCane* offers timely navigational guidance to the visually impaired.
- *iCane* issues warnings of specific road conditions (e.g. crossroad etc.) ahead.

- *iCane* helps locate nearby services, e.g., shops and government offices.
- *iCane* maintains a personal map to facilitate revisiting points of interest.

The rich information and interaction offered by *iCane* assists the visually impaired to go beyond the boundaries of their homes and familiar environments. *iCane* is a first step toward an obstacle-free space for the needed to help explore and broaden his/her experience and enjoy life in a brand new way.

## 2 System Overview

The basic design idea is to embed the common white cane with a location sensor. In particular, *iCane* is equipped with an RFID reader, which connects to a Personal Digital Assistant (PDA) with Bluetooth earphones. RFID tags preloaded with information are adequately placed throughout the public space. After sensing the RFID tags on the ground, the RFID reader embedded in *iCane* sends the information to the connected PDA to check for warnings (e.g. crossing points, hazard etc.) as well as additional information of interest, e.g. a virtual signboard for a nearby shop or the menu for a restaurant straight ahead. This way the visually impaired can stroll around the streets freely. In addition, given a destination, the map system is able to compute the best direction to go based on the target location. The instructions are then transmitted to the user via speech through the Bluetooth earphones. A microphone is adopted as the input device, and voice commands from the user are processed through speech recognition software. In summary, *iCane* is designed to provide rich environmental information to the visual impaired through a spoken language interface.

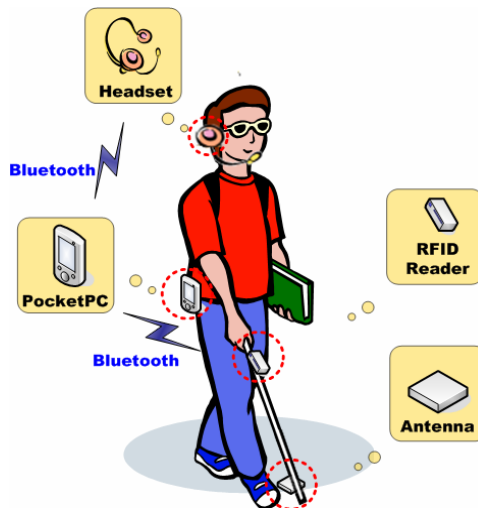


Fig. 1. The *iCane* System Overview

### 3 Hardware Implementation

Since the visually impaired use white canes frequently, designing white canes must be done with care. There are several issues which must be considered. First of all, the weight of the cane cannot be too heavy since the visually impaired needs to use the cane over a long distance. Second, the user should be able to carry the white cane conveniently thus it should be foldable to reduce size. Third, since the cane is designed for daily use, it must be durable. Last, the cane should be able to extend tactility of our visually impaired users. However, there are tradeoffs between *iCane* and original white cane. For example, to increase the functionality of original white cane, we might place sensors on it and thus increase the weight. Currently we adopt an existing white cane and attach our sensor and antenna to it. In order to increase the accuracy of detecting RFID tags, we place the antenna on the bottom of white cane to sense tags on the ground. For preventing damage from the environment, such as rain and accident collision, the reader is located on the handle near their hands and connected to the antenna by extended wire coiled around the body of white cane. All these components are protected in waterproof pads.



**Fig. 2.** On the left hand side of the figure is the *iCane* overview; on the top right of the figure is our RFID antenna; on the bottom right of the figure is our RFID reader

**Location Sensor.** There are several existing locating systems and one of them is GPS, which has been used for many years. But the typical error of 6 to 12 meters is not accurate enough for the visually impaired. To achieve high accuracy using GPS, the cost becomes prohibitively high. Besides, indoor GPS doesn't perform well. An alternative locating system is the WiFi-based positioning. To make it accurate, it also requires several access points placed in a small region. For this reason WiFi technology is much more suitable in the indoor environment.

Due to the limitations in the GPS and WiFi-based positioning, we decided to adopt RFID technology. There are some advantages of adopting RFID. First, the accuracy depends on how dense we pave the tags. And the price of a passive tag is relatively low – this can reduce the cost of building infrastructure. Second, it can work in both indoor and outdoor environment, which expands the service area of the *iCane*.

RFID receiver contains RFID reader and antenna. We adopt SkyeTek EA1 to be the antenna and the RFID reader module we use is a GigaTek's product (RWM600). The RFID reader supports RS232 protocol for us to combine it with an RS232 – Blue-

tooth adapter. Thus the communication link between PDA and RFID reader is through Bluetooth wireless channel. The RFID reader uses a 9V battery as power supply and can operate continuously about 6 hours.

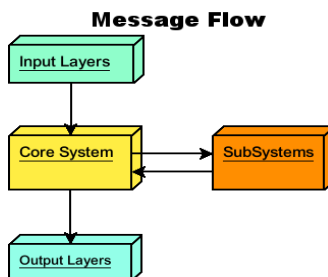
**Computing Device.** We choose the PDA as our computing device. The HP iPaq 5550 PocketPC is selected as the platform for *iCane*. It is connected to the RFID reader and the headset via Bluetooth technology.

**Input and Output Device.** We provide input and output devices for the user to communicate with the *iCane*. First, to provide an accessible interface for the visually impaired, we choose the Bluetooth headset as our user interface. This enables the users to talk to the *iCane* directly, making the application intuitive. The users can also receive instructions from *iCane* through voice guidance. Second, Sometimes people may be in a quiet place where they don't want to make noises; or they may not want others to hear where they want to go. So we provide the buttons on PDA as another input device.

## 4 Software Implementation

### 4.1 System Architecture

We design the system following the object-oriented paradigm. The system is divided into 4 layers: input, core system, subsystems, and output. Each object of the 4 layers is responsible to its own work. And except the core system, other objects can be substituted by dummy objects. This kind of architecture makes us able to build a complete system even in initial developing period, and we can make sure that the workflow of the system is correct. In the following stages, we replace these dummy objects by implemented components one by one.



**Fig. 3.** Software Message Flow

The functionality of each layer is described below:

- **Input Layers:** Communicate with input devices. Wrap input data into *InputData* structure and forward them to the Core System. Input Layer contains LocationSensor, speech recognizer, hardware buttons.

- **Core System:** Forward data from Input Layer to subsystems. Wrap computed data into *OutputData* structure and send them to adequate output layer component.
- **Subsystems:** Currently there are two subsystems, and their responsibilities are listed below.
  - *User Tracker:* Track the recent user’s position history and predict the user’s position.
  - *Map System:* Store map data of the real world. Provide services such as path planning.
- **Output Layers:** Provide output ability to the system. Make the system able to transfer messages to users. Output Layer contains Text-To-Speech, text logger.

## 4.2 Software Components

Implementation details of primary components are described as follow:

**LocationSenser: GigatekReader Adapter.** We use the GigaTek RFID Reader as the main location sensor in *iCane*. The class *GigatekReader* implements the protocol that communicates with the GigaTek RFID Reader. Each time reader’s tag-in event occurs, the *GigatekReader* sends out commands according to the requirements to read contents in tags. The tag ID is then converted to the coordinates by *TagDB*. The coordinates and data is wrapped into *InputData* structure and is put into the internal queue for the *CoreSystem* to take out later on.

**TagDB.** Tag Database stores data of all tags in the environment and provides an interface which enables others to query tag ID and its corresponding coordinates. In our first design, we stored the real world coordinates in each tag and read them out when needed (e.g., intersection, corner), and the system gave instructions that tell the user where to go. However, it takes extra time to read out the content written in tags, and the way the visually impaired use their cane may cause them to lose important information, such as intersection. According to the experiments we have done, even if we placed 3 Blister paving (15 tags per 30\*30 brick) contiguously, the missing rate is still up to 20% at a relatively slow walking speed (1.5km/hr). This made us change our design. We considered storing all mapping data including tag ID and its corresponding coordinates in the PDA. Taking the map of Taipei City (where the streets are highly concentrated) for example in 2003, the total length of all streets in Taipei City was 1,536,713 meters. Suppose that we place one tactile paving every 30cm, and the data of one tactile paving consist of tag ID (8 bytes) and its coordinates (8bytes), so the total size needed for one tactile paving is 16 bytes. We also considered that there are two pavements, one on each side of the street. We are able to store the map data of all streets in Taipei City within  $16 \times (1,536,713 / 30) \times 2$  bytes (about 156MB). Consider today’s storage devices, this size is quite acceptable. So we decided to store the whole mapping data of a city in the Tag Database.

As for implementation, because the size of RAM in PDA is limited, the whole mapping data cannot be loaded into memory. So we sort the mapping data according to tag ID in a file. TagDB class does binary search in the file to find out the corresponding coordinates in the real world.

**Tag Encoding.** There are several kinds of tactile paving with different functionality. Directional paving and Blister paving are most commonly seen. In our system, we distinguish the two by tag ID. The reason why we do this is because the tag-in event occurs immediately after the RFID reader senses a tag, and it will read in tag ID at the same time. But if we want to read the data stored in tags' memory, it requires additional command such as select ID and read ID, which requires additional time. And reading data from tags cannot be done when the visually impaired swing their canes fast. In order not to slow down user's walking speed, tags are classified into two types. By encoding the tag ID, we can decide whether to read the data in tags just after tags are sensed.

**Table 1.** Tag Encoding Table. Tags are classified by the Least Significant Nibble (LSN) of tag ID and there are 10 types of tags

LSN	Function
**00	Directional paving
**10	Virtual Signboard
0001	Blister paving - intersection
0011	Blister paving – corner
0101	Blister paving – stairs
0111	Blister paving - escalator
1001	Blister paving – elevator
1011	Blister paving – ascending road
1101	Blister paving – descending road
1111	Blister paving - obstacle

**CoreSystem.** The *CoreSystem* examines whether the tag-in event has happened by polling the RFID reader object 20 times per second. If there is a tag-in event, The *CoreSystem* takes out the *InputData* and forwards it to *UserTracker* and *MapSystem* for further computation. It also reads out the content of a virtual signboard. And the computed data is wrapped into *OutputData* structure and sent to Output Layer.

#### *i. MapSystem*

Our map system deals with two main tasks. First, it maintains the real world's map and other environmental information. Second, it provides path-planning service that help users in navigation.

In the first part, we map addresses and other location names into geographical coordinates. We provide a query interfaces makes possible the conversion between human readable names and absolute coordinates. Our map structure is simply defined as follow: pairs of coordinates represent roads in the real world, and the map is constructed by all those roads; and a database which contains the mapping between addresses and coordinates is also maintained.

For path planning, given two locations (current location and destination), our map system finds out the shortest path for the user. Since the entire map data is stored in the system, coordinates of the two locations are available. We can than achieve path

finding by the A\* search algorithm, which narrows the search area with a heuristic function. Once the user sets the destination, the map system finds out the shortest path from current location. And then the map system will keep monitoring if the user is on this shortest path. If the user moves out of the planned path, the map system automatically computes a new shortest path immediately.

### ii. *UserTracker*

The *UserTracker* consists of a location history which records the recent location information of the user. We use the location history to approximate the user's walking speed and predict the current location of the user. If the user is closing to the Blister paving, The *UserTracker* can remind the user to slow down for getting important road condition information.

**User Interface (Speech Recognizer, PDA Buttons, TTS).** The user interface consists of two parts: input and output. The output is presented in the form of speech, while the input part consists of speech and PDA buttons. The interaction between *iCane* and users can be accomplished by speech, buttons or both.

In order to achieve an easy-to-use interface, the input method through PDA control buttons is designed with simplicity in mind. There are two direction buttons (for menu selection) and two command buttons (for confirm and cancel). It functions like a cell phone when you input through PDA buttons. You can choose the function with direction buttons and then press the confirm button to execute it.

Also, speech interface is provided. Through speech recognition, user can read out specific words to act out the desired function without pressing any buttons. To process speech recognition, we use the NeuVoice Audient Pocket PC as the developing SDK. As for the output, all output messages including virtual signboard, prompting messages, and instruction messages are delivered to the user by the Text-To-Speech module.

## 5 Experiment Results

To assure that every component works correctly, unit test class for each main component is built. Besides, test cases which simulate real environment are also built through RFID emulator. By using the Text Logger, we could examine that if the system output is the same as we have anticipated.

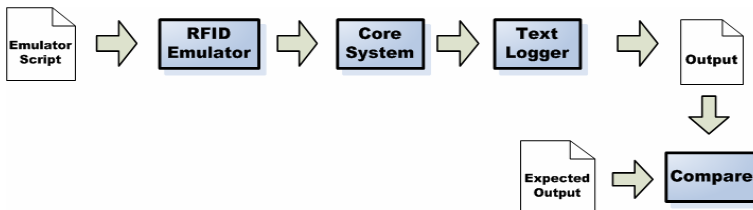
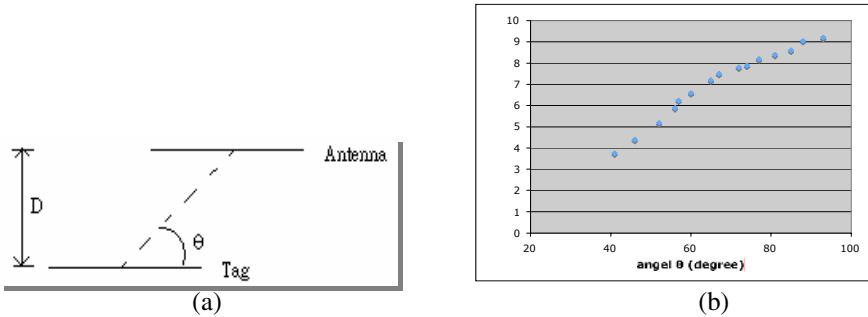


Fig. 4. Software Testing Automation

**RFID Emulator.** In order to let the system work without real RFID reader in early development, we designed an RFID emulator which simulates behaviors of real RFID reader. The benefit is that we can build up a complete system that is workable in a short time. Besides, test scripts which simulate real environment could be designed through the emulator, and automatic testing to the whole system could be done.

**Assistant Tool for Paving Tags.** To support infrastructure, a large number of tags need to be paved in the environment, and the tag ID and its corresponding coordinates should also be recorded. To construct a simulating environment efficiently, an assistant tool has also been developed. The assistant tool simply asks the RFID reader to read in tags on the street sequentially, automatically computes the corresponding coordinates of each tag, and saves them in the Tag Database. With the aid of this tool, we can place the RFID reader on a cart. After walking along all the streets with the cart, the mapping of tag ID and coordinates is then computed.

**RFID Sensing Profile.** The sensing range of RFID technology is limited, and varies with the angel between the antenna and the line connecting the antenna and the tag. We define  $\theta$  to be the angle between the line connecting the centers of the tags and the antenna, and  $D$  the vertical distance between the centers of the tags and the antenna, as shown in Fig. 5(a).



**Fig. 5.** (a) Definition of  $D$  and  $\theta$  (b) Detectable Distance and Angel

If we measure  $D$  under various  $\theta$ , we can get their relationship as shown in Fig. 5(b).  $D$  increases as  $\theta$  increase and reaches the maximum when  $\theta = 90^\circ$ .

As the above experiment shows, the tag should be near by the antenna when the cane contacts the ground (no longer then 5cm) in order to be sensed. Besides, the length of the tag is 9.6cm, so we decide to pave tags every 15cm to ensure that there will be at least one tag in the sensing range if we walk along the tactile paving.

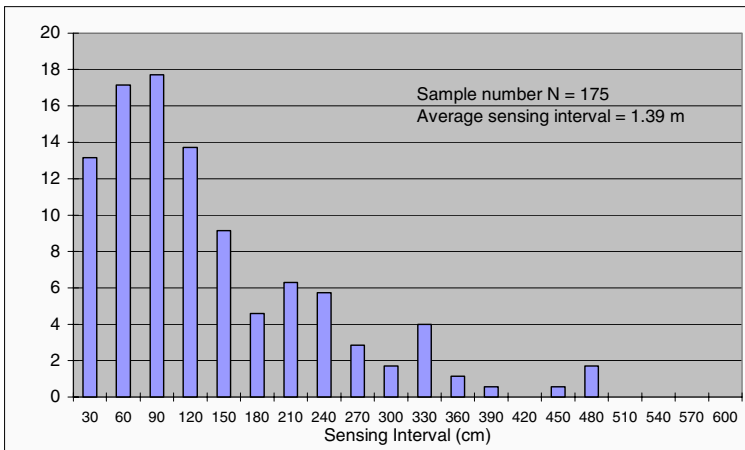
**iCane Sensing Rate.** The main factor that will affect the reliability of our system is the sensing rate of the tag. In order to assure data correctness, the location information is expected to be updated as often as possible, i.e., the distance between each tag being sensed should be as short as possible.

On testing the sensing rate, two issues should be taken into consideration. First, the way users wave their canes. Second, the speed at which the user walks. In the first



case, the visually impaired needs be taught how to wave a cane in three gestures. In our tests, we elect just one of the waving gestures: knocking at the ground in the walking path of the user and waving high by two sides to detect the obstacles. And the walking speed is fixed at 3 km/hr – about the walking speed of an adult.

We have done some experiments to measure the sensing interval and analyze the results. Our experiment environment was set as a 10 meter-long tactile paving assembling by 30 cm x 30 cm movable paving. The experiment procedure is, first, the user setting up with *iCane* system walked through the tactile paving as stable as possible and we recorded the interval between the detected paving. Then, we repeated this experiment for 40 times and established a figure showing probability distribution of the sensing interval. The result shows that the average sensing interval is about 1.39 meters. In most of the case (80%), users can sense the next tag in a reasonable interval - less than 2 meters. Therefore, this result confirms the feasibility of *iCane* system.



**Fig. 6.** Distribution of the Sensing Interval

Software prediction is used to improve the reliability of the *iCane* System. The software prediction method approximates the user’s walking speed using previously sensed tag locations in the tag history. If the user is closing to the Blister paving, he will be notified to slow down his walking speed.

## 6 Conclusion

Using *iCane*, visual information about the public space now becomes available for the visually impaired. With the help of RFID technology, we are able to retrieve richer information and present it in more accessible forms, e.g. voice, to the sightless user. In addition, location-based services such as path finding are provided to help the visually impaired in unfamiliar environments.

The *iCane* system will become more feasible with the maturity of the RFID technology. It is a first step toward an obstacle-free space for everyone to explore and

broaden his/her experience and enjoy life in a brand new way. The visually impaired is no longer confined within the limits of their homes and familiar environments. The rich information and interaction offered by *iCane* open up many new possibilities and allows the visually impaired to go beyond the boundaries of their disability.

## Acknowledgements

This research is supported in part by a grant (NSC-93-2218-E-002-148) from the National Science Council in Taiwan and a grant from Intel.

We would like to thank Jane Hsu, our advisor, John Wang, who made revision in our first draft paper and Cultural & Educational Foundation for Blind, who gave us comments during our research.

## References

1. "RFID Journal." [Online] Available <http://www.rfidjournal.com/>, 2005.
2. Klaus Finkenzeller. RFID Handbook : Fundamentals and Applications in Contactless Smart Cards and Identification, Second Edition. John Wiley & Sons, Inc., 2003.
3. Stuart Russell, Peter Norvig. Artificial Intelligence: A Modern Approach, Second Edition. Pearson Hall, pp. 97-110, 2003.
4. "Speech Recognition SDK for Windows" [Online] Available <http://www.neuvoice.com/products/audientppc.php>, 2005.
5. Iwan Ulrich, Johann Borenstein. "The GuideCane-Appling Mobile Robot Technologies to Assist Visually Impaired.", IEEE Transactions on Systems, Man, and Cybernetics, —Part A: Systems and Humans, Vol. 31, No. 2, pp. 131-136, March 2001.
6. Vladimir Kulyukin, Chaitanya Gharpure, Nathan De Graw, John Nicholson, Sachin Pavithran. "A Robotic Guide for the Visually Impaired in Indoor Environments." Proceedings of the Sixteenth Innovative Applications of Artificial Intelligence Conference (IAAI-04), San Jose, CA, 2004.
7. Tomohiro Amemiya, Jun Yamashita, Koichi Hirota, Michitaka Hirose, "Development of Virtual Leading Blocks for the Blind", In Proc. of VRSJ 8th Annual Conf., pp. 359-362, VRSJ 8th Annual Conf., Gifu, 2003.9.(In Japanese)
8. "PocketPC (WinCE) GPS and Navigation." [Online] Available <http://www.gpsinformation.org/dale/PocketPC/wince.htm>, 2005.
9. "MapServer Homepage." [Online] Available <http://mapserver.gis.umn.edu/index.html>, 2005.

# **ORaid: An Intelligent and Fault-Tolerant Object Storage Device\***

Dan Feng\*\*, Lingfang Zeng, Fang Wang, and Shunda Zhang

Key Laboratory of Data Storage System, Ministry of Education,  
School of Computer, Huazhong University of Science and Technology, Wuhan, China  
dfeng@hust.edu.cn, zenglingfang@tom.com

**Abstract.** Hints for traditional storage system come from three aspects: file or directory attribute layout, user input and file content analysis. But in the OBS (object-based storage), object, a new fundamental storage component different from traditional storage unit (file or block), provides ample hints for storage system, which are help for designing more intelligent (or smarter) storage device. RAID (redundant arrays of independent disks) is a popular mechanism to offer fault-tolerance storage. But RAID based on file or block interface has very limited intelligence. This paper presents a novel object-based device: ORaid (object RAID). ORaid consolidates disk space of individual disk into a single storage pool and implements object-sharing and fault-tolerance through the object-interface. In ORaid, storage object is stored in form of original block fragments and their XOR verification fragments are among the disks of the device. As the higher abstract of data encapsulation, storage object provides more intelligent properties, and ORaid has more effective ability to implement online data re-layout and online capacity expansion.

## **1 Introduction**

With exponentially increasing information, storage research community is quite aware of the management problem of storage system. Enhancing storage device intelligence, specially self-managing and self-configuring, has been well-publicized calls to action. Examples include the recovery oriented computing, the Self-\* Storage [1], University of California Santa Barbara's Active disks [2], CMU Active disks [3]. University of California at Berkeley's IDISK [4] offloads application-specific functions, such as database scanning, in order to reduce server processing load. SDDS [5] discusses mechanisms for detecting and recovering from non-recoverable read errors and drive failures, introducing improved techniques for detecting errors in disk reads and fast

---

\* This work was supported by the National Basic Research Program of China (973 Program) under Grant No. 2004CB318201, Huo Yingdong Education Foundation under Grant No.91068, the National Science Foundation of China under Grant No.60273074 and No.60303032.

\*\* Corresponding author.

recovery from disk failure. ISTORE [6] achieves self-maintenance via hardware platform with integrated diagnostic support, reactive self-maintenance by a layered, policy-driven runtime system that provides a framework for monitoring and reaction, proactive self-maintenance with support for continuous on-line self-testing and component characterization and a standard API for interfacing applications to the runtime system. OceanStore [8] is a global persistent data store designed to scale to billions of users. It is designed to have a very long MTDL, but at the cost of dramatically increasing the number of disk requests per block written.

Above mentioned are typical smart storage device. Local intelligence is achieved in those devices and is a basis of the whole storage system. In addition, RAID [9], which has been developing for approximately twenty years, shows itself in some forms: software RAID [20], defined as a system that distributes data redundantly across an array of disks attached to each of the workstations connected on a high-speed network, provides higher throughput and availability. HP AutORAID [7] is attribute-managed storage and a two-level storage hierarchy implemented inside a single disk array controller. Reference [10] has proposed iSCSI RAID (or iRAID for short) to improve iSCSI performance from the iSCSI target point of view. iRAID provides a direct and immediate solution to boost iSCSI performance and improve reliability. Specially, ORAID, as an intelligent and fault-tolerant object storage device, is introduced in this paper.

The rest of this paper is organized as follows: Section 2 describes the background and our motivation. Section 3 studies those characteristics of storage object, such as object attribute, object method and object-based storage policy. Section 4 provides the implementation and typical application of ORAID. And the conclusions and the future works are given in section 5.

## 2 Background and Motivation

Indeed, in the past 20 years, the use of technology improvements to achieve lower cost, as well as increased performance, has been a major theme in the storage industry. For data storage, OBS (object-based storage) is the next wave of storage technology and devices [13]. Smart storage system is very important for emerging data-intensive network applications. An entire industry and research community has emerged to achieve fault-tolerance and smarter storage by many ways: (1) Modifying file system. (2) Smarting storage device. (3) Enhancing network fabric intelligence. (4) Considering overall storage system. All of these endeavors have been productive, but they cannot help industry to bridge the gap between intelligence storage system and real business value when they get insufficient hints from file or block.

Security and performance of aggregate bandwidth are emphasized in NASD project [12] and Lustre cluster file system [16], respectively. The function of RAID is mainly realized by data layout policy of the metadata server, the same to the iRAID.

It may be a new approach for us to design smart and fault-tolerance storage device with the help of object storage. This principle has been considered for many years in the storage system community. But traditional storage systems based on files or blocks, such as DAS, NAS and SAN, cannot provide more accurate or ampler hints. But, objects in OBS already give useful hints to the storage system, in the form of object

OID and other attributes, and that storage system can successfully get smarter from these hints. Also, with those hints, RAID-based storage device may change its RAID running level (online re-layout works) or online capacity expansion. Hints about an object's access pattern, size, and lifespan can aid in a variety of ways including improving the object's layout on RAID and increasing the effectiveness of managing and caching [17].

### 3 Overview of Storage Object

Objects are of variable size and can be used to store entire data structures, such as files, database tables, medical images, or multimedia. Object data is application data and include some attributes (size, create time, object type, lifespan, ownership etc.) [11], [12], [15], [16]. The object is the fundamental unit of data storage in OBS. Also, object is a logical collection of bytes with well-known methods for access, attributes describing characteristics of the data, and some policies that realize self-management.

#### 3.1 Storage Object Attribute Is the Foundation Stone of Hints

Object represents an abstract higher-level. A storage object is variable-length and can be used to store any type of data. Thus different type of data has different attributes. If we use a uniform style to contain attributes for any type of object, more disk space may be wasted while saving objects with their attributes in disks. So we propose a scalable way to present attribute, which we call attribute card. The attributes in the same attribute card have close relationship. An attribute card can be identified by an attribute card ID (AID), and an attribute within the attribute card is indexed by an attribute index (AIdx), so an attribute is uniquely presented as a pair (AID, AIdx).

We define a common attribute card, which contains attributes that every object possesses. For example, any object has attributes such as create time, size, last modified time et al, all of which can be included in the common attribute card. Any new type of attribute card can be defined according to applications, and such attribute card is application-defined attribute card. Considering attributes of a multimedia object, a multimedia attribute card can be defined as follow: QoS, frame rate, video size et al. However, a database record attribute card may be composed of attributes as follow: record count, record format et al.

In OBS, ORaid, an object-based storage device, is also a special device object and comprises some attributes, such as ORaid's initial capacity, remaining capacity and IP address. For convenient managing, other object attribute can be extended to three types, such as public attribute, privacy attribute, share attribute. Share attribute (we want to make a difference between access attribute and access pattern) contains information about its environment, group and user access control information. E.g., share attribute provides greater interoperable access to objects and provide information to ORaid that describes the meaning of the objects and facilitates more effective data management, such as on-disk layout and resource provisioning. Privacy attribute (similar to storage attribute [4]) can tell the storage system how to deal with an object, for instance the level of ORaid to apply, the size of the capacity quota or the performance parameters

required for that data. Once ORAID accesses to attributes describing the objects, it can use these attributes to achieve improvements in the device. E.g., ORAID might make sure that the hot spot object is available more quickly, or that previous versions of objects are automatically saved when it detects that objects are being changed. When ORAID understands the meaning of data it contains, it can use that information to create new storage applications that take advantage of ORAID ample processing power. In addition, information such as backup or QoS (quality of service) requirements, object access patterns (sequential or random), or relationships to other objects, is also stored as share attributes.

Traditional systems can automatically learn to classify the properties of files and predict the properties of new files by exploiting the strong associations between a file's properties and the names and attributes assigned to it [18], [19]. ORAID analyses object pattern from the following three aspects: (1) Object attribute analysis, (2) Access-based object attribute analysis (including application assistance and existing user input), (3) Inter-object relationships. However, we have to emphasize that not only can ORAID get those hints (most of them are local), but also metadata server can attain some hints from global object metadata information.

### **3.2 Storage Object Method Is the Executive Unit**

Traditionally, a block-based disk can only perform READ and WRITE operations on data. Object-based device changes this way by allowing users to upload application-specific operations at storage device and performing operations on data while receiving requests from clients. Furthermore, object-based device must provide more flexible and scalable operations and can be applied to more application fields. OBS provides high management by extending object method.

Each object in an ORAID also has associated with read/write streams. Users can insert any modular methods into read and write streams. Within a stream, methods are connected one to the next to form a method chain. When users read/write objects, the objects enter the stream at one end, progress through the method chain, and exit at the other end. By the form of method chain, object method is executed when the data passing through the chain. ORAID provides flexible methods by supporting arbitrary data stream operations. Thus, users can upload any kind of methods, and build any kind of method chains for any objects.

For ORAID, users can associate an object with specific method chains in two ways: transient way and persistent way. In transient way, method chains can be built when an object is opened and destroyed when the object is closed. In persistent way, method chains for an object can be created or destroyed through a register/un-register request to ORAID. After an object is associated with its method chains through registration, the operations in the method chain are always implicitly performed.

### **3.3 Storage Object Policy Is the Brains of ORAID**

Object storage policy management is rule-based scripts that identify ORAID conditions, states, and events, and generate appropriate actions. In the network storage world, it easily tracks and acts upon object storage network devices (e.g. ORAID) and

infrastructures. Because of the scalability of object-based storage, policy developers are easy-pressed to code policies that can run in object storage environments. There are so many storage realities to observe, such as runaway storage data, greedy database applications that ruthlessly grab disk space. Within this demanding environment, OBS use policies to help them meet service-level agreements (SLAs), keeping object storage devices provisioned and configured so that they can meet required space, performance, and availability metrics.

SNIA [22] defines storage-related policies as “the measurable, enforceable, and realizable specification of methods, action and/or desired states that meet service requirements in a storage-based information infrastructure.” This means that policies are machine-based reactions that deploy in response to changes in the object storage environment. These changes might be driven by events, conditions, or computing processes. Users can measure and enforce both changes and policy-generated reactions from a user interface.

Object storage policies may not be as mature as network storage policies, but they are important additions to the storage management thoughts. Object storage policy can manage object storage devices to improve and automate backup-and-restore and archiving procedures, supply bandwidth to demanding applications, and assure that a critical backup has the resources it needs.

In our ORAIID, rules and policies are separated. It is because that one policy may correspond to several rules, and one rule may adapt to different policies. The rule pool is filled with general values regarded to be useful by one or more of the management policies. Most popular rules such as time, recency of access [15], frequency of access, capacity and size are initially registered to the pool. A new policy registered into the policy pool has to contact corresponding rule in case they are not already registered. So ORAIID can provide a solution for dynamic loading or unloading policy.

Also, for ORAIID, those policies themselves are just descriptions of how to help OBS implement system management function and specify system local states and how to response to them. These may include any proposed storage system management policy because some state information may be exchange among ORAIIDs. When clients or application services operate objects, metadata server (has global information) and ORAIID (has local information) set correlative object attribute values. By getting object attributes and statistic values, OBS will ensure that the relative policy for the current workload or performance is triggered<sup>1</sup> (in ORAIID) and therefore have the largest effect on the storage system management decisions.

## 4 Design, Implementation and Application of ORAIID

### 4.1 Functions of ORAIID and Its Metadata Server

Figure 1(a) shows those function modules in ORAIID. With SCSI subsystem and RAID driver, the lower disk array realizes those functions of traditional hardware RAID.

---

<sup>1</sup> OBS triggers the policy depending on the compare between object attribute values and rules from the rule pool.

Moreover, in the middle layer of figure 1(a), the storage layout frame provides online re-layout and online capacity expansion. In fact, the function of online re-layout includes the RAID level management (or configure). All above mentioned implementations all require the support of logic volume management which is realized by the logical volume manager (LVM) [21]. The logic volume management is used for online disk storage management. The logic volume management considers all installed disks as pools of data storage and provides easy-to-use online disk storage management for computing environments.

With those online self-managing functions, the management of ORAID does not require that machines be taken off-line at a major convenience to users. So, in the distributed client/server environment, databases and other resources maintain high availability in ORAID. ORAID is easy to access, improves performance and ensures data availability and integrity. However, if we want ORAID to carry out those function without manual intervention, different from traditional RAID, ORAID have to attain some intelligence. Fortunately, with those help of object interface in figure 1(a) and object storage service in figure 1(b), storage objects in ORAID can present adequate hints. Figure 1 also shows that metadata server and ORAID provide storage services for clients (or users) by high-speed TCP/IP network.

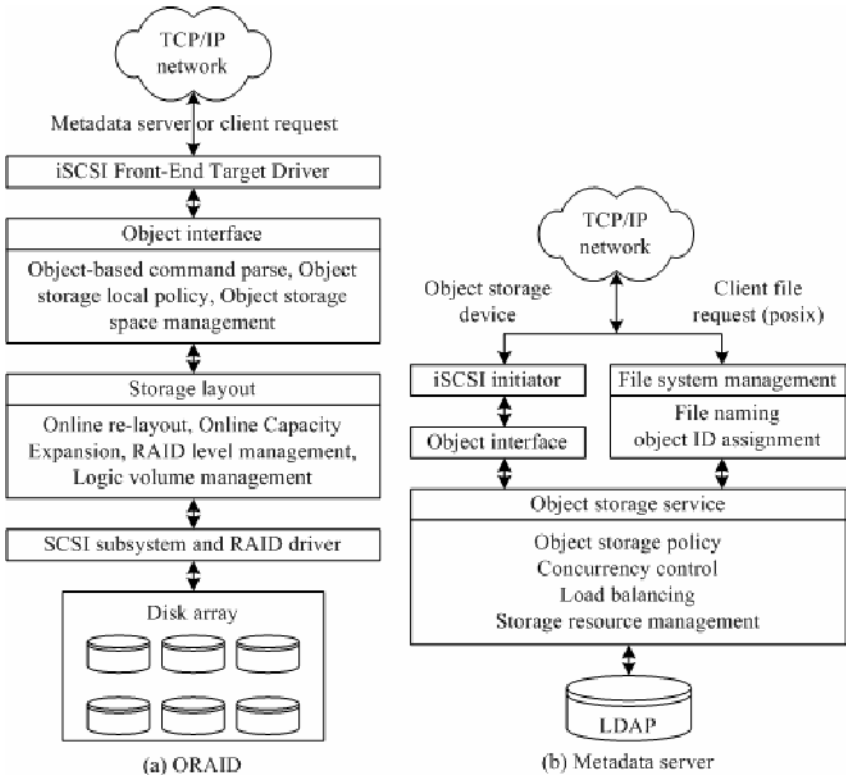


Fig. 1. Function modules of the ORAID and the metadata server



### 4.2 Program Implementation of Metadata Server

Flow chart of program in metadata server is showed in figure 2. In metadata server, most of those requests are write/read service of clients (or users). Other services of clients include permission, rm, mkdir, ls etc. In fact, the security of object access also is very important, but, it is not the key objective of this paper. Figure 2 does not show the service of object storage policy. In the following section 4.3, we will discuss this together with storage local policy in ORaid device.

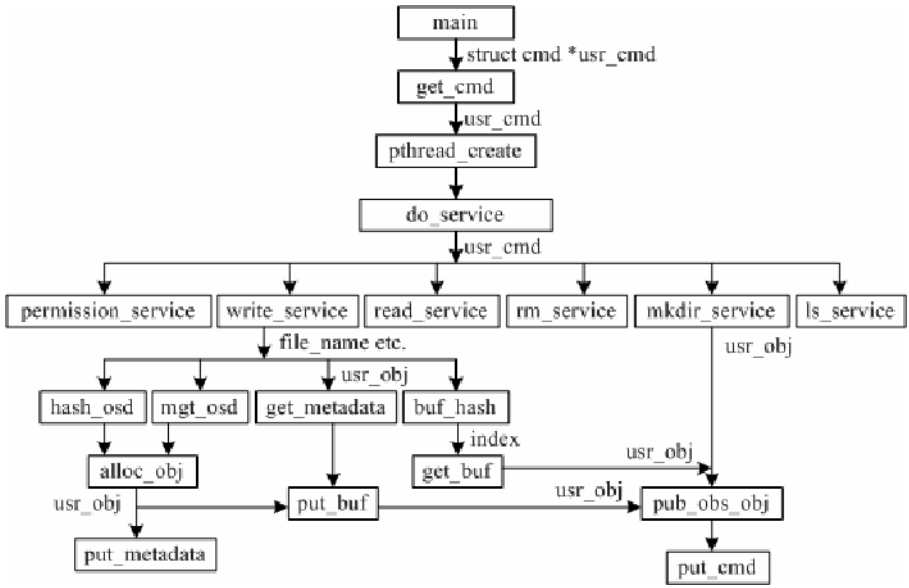


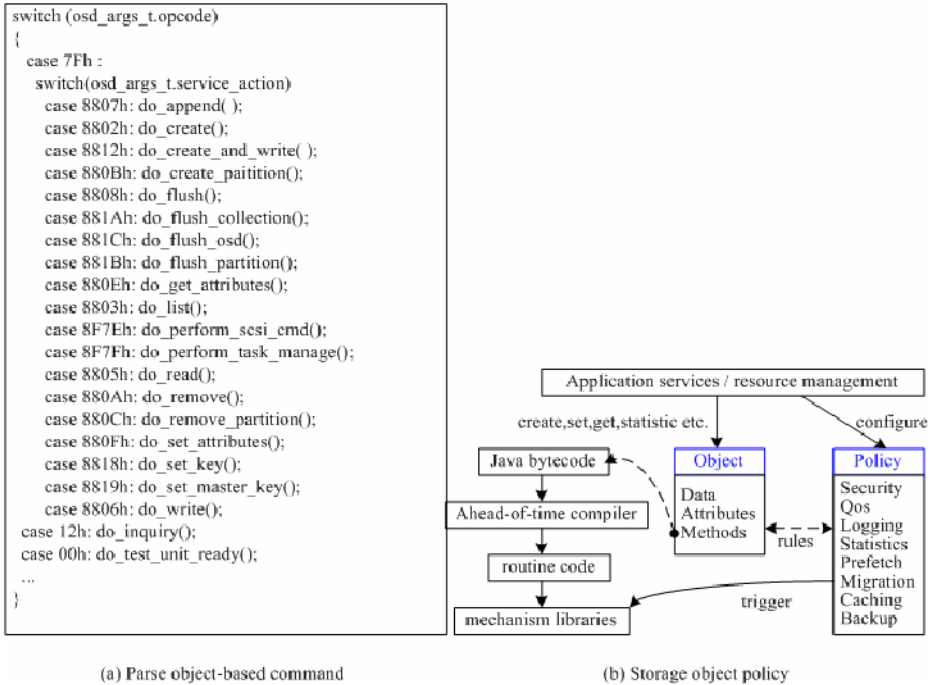
Fig. 2. Flow chart of program in metadata server

### 4.3 Implementation of Object Command and Local Policy in ORaid

As an object storage device, ORaid must implement object-based storage device commands [23]. Figure 3(a) provides the parse program for object-based command in our ORaid, and this implementation is based on Intel Lab's software reference implementation of iSCSI [24].

The processing flow of storage object policy is showed in figure 3(b). In ORaid, object and policy are operated by application services or resource management. Besides some system methods, object also has some itself methods. Obviously, object attributes are set some values, or storage subsystem can get some statistic information by object attributes (Of course, both ORaid and metadata sever record some access information when objects are operated by application services.). By comparing rules, some policies may be triggered in ORaid and some adaptive code in mechanism library may be executed according to those policies.

With those hints provided by storage objects, ORAID usually changes the storage layout to change the redundancy or performance characteristics of the storage. The logic volume management adds redundancy to storage either by duplicating the address space (e.g. mirroring) or by adding parity (e.g. RAID-5). Performance characteristics of storage can be changed by changing the striping parameters which are the number of disks and the stripe width.



**Fig. 3.** Implementation of object-based command and storage object policy in ORAID

ORAID makes changing fault-tolerant storage configuration easy, without requiring a data backup and restore cycle. If ORAID composes of hot-pluggable disks (e.g. SATA disk), storage expansion or extension can be performed online -- without shutting down the server operating system. Online capacity expansion adds storage capacity to an array. If an existing array is nearly full of data, simply by adding a new physical disk, ORAID can expand the capacity without disturbing the existing data. According to those storage object policies, ORAID automatically sets the disk hardware and configuration.

In addition, in object-based storage system, above mentioned works are cooperatively finished between ORAID and metadata server. For instance, ORAID must send some state message to metadata server when it is configuring its layout. Also, metadata server must record capacity expansion of ORAID.

## 5 Conclusions

As business trends evolve and new technologies emerge, storage system is becoming more complex, compounding the challenge of sustaining the intelligence that storage community seeks. This paper has shown that object attributes are strong hints of how that object will be accessed and managed in a smart object-based storage system. Then storage object method and policy are introduced. Furthermore, a novel intelligent and distributed fault-tolerance storage mechanism called ORAIID is provided. Using those characteristics of storage object in OBS, ORAIID realizes fault-tolerance storage, online re-layout and online capacity expansion.

## References

1. John D. Strunk, and Gregory R. Ganger. A Human Organization Analogy for Self-\* Systems. First Workshop on Algorithms and Architectures for Self-Managing Systems, in conjunction with Federated Computing Research Conference (FCRC). San Diego, CA, June 2003.
2. A.Acharya, M.Uysal, and J.Saltz. Active disks: programming model, algorithms and evaluation. Proceedings of the 8th Conference on Architectural Support for Programming Languages and Operating System (ASPLOS VIII), pp. 81-91, Oct. 1998.
3. E. Riedel, C. Faloutsos, G.A. Gibson, and D. Nagle. Active disks for large-scale data processing. *Computer*, Vol 34, Issue: 6, pp. 68-74, June 2001.
4. K.Keeton, D.A.Patterson, and J.M.Hellerstein. A case for intelligent disks (IDISks). *SIGMOD Record*, 27(3), Sept. 1998, available at <http://www.acm.org/sigmod/record>
5. Qin Xin, E.L.Miller, T. Schwarz, D.D.E. Long, etc. Reliability Mechanisms for Very Large Storage Systems. Proceedings of 20th IEEE/11th NASA Goddard Conference on Mass Storage Systems and Technologies, pp. 146-156, April 2003.
6. A.Brown, D. Oppenheimer, K. Keeton, R. Thomas, J. Kubiatiowicz, and D.A. Patterson. ISTORE: Introspective Storage for Data-Intensive Network Services. Proceedings of the 7th Workshop on Hot Topics in Operating Systems (HotOS-VII), Rio Rico, Arizona, March 1999.
7. John Wilkes, Richard Golding, Carl Staelin, and Tim Sullivan. The HP AutORAIID Hierarchical Storage System. In *High Performance Mass Storage and Parallel I/O: Technologies and Applications*, IEEE Computer Society Press and Wiley, pp. 90-106, 2001.
8. Chris Wells. The OceanStore Archive: Goals, Structures, and Self-Repair. March 2004, available at <http://oceanstore.cs.berkeley.edu/publications/index.html>
9. D.A. Patterson, et al. A Case for Redundant Arrays of Inexpensive Disks (RAID). *ACM International Conference on Management of Data (SIGMOD)*, pp: 109-116, 1988. QUID Web Proxy Cache, available at <http://www.squid-cache.org>
10. Xubin He, Praveen Beedanagari and Dan Zhou. Performance Evaluation of Distributed iSCSI RAID. International Workshop on Storage Network Architecture and Parallel I/Os. New Orleans. September 2003.
11. Butler W. Lampson. Hints for Computer System Design. In *ACM Operating Systems Review*, volume 15(5), pp. 33-48, October 1983.
12. Thomas E. Anderson, Michael D. Dahlin, Jeanna M. Neefe, David A. Patterson, Drew S. Roselli, and Randolph Y. Wang. Serverless network file systems. *ACM Transactions on Computer Systems (TOCS)*, Vol 14, Issue: 1, pp. 41-79, February 1996.
13. P. J. Braam. The Lustre storage architecture. Technical report, Cluster File Systems, Inc., January 2004, available at <http://www.lustre.org/docs/lustre.pdf>

14. SNIA, Object-Based Storage Devices (OSD) workgroup, January 2004, accessible from <http://www.snia.org/osd>
15. M. Mesnier, G.R. Ganger, and E. Riedel. Object-based storage. *Communications Magazine, IEEE*, Vol 41, Issue: 8, pp. 84–90, Aug. 2003.
16. Intel Corporation. Object-Based Storage: The Next Wave of Storage Technology and Devices. January 2004, accessible from <http://www.intel.com/labs/storage/osd/>
17. Ke Zhou, Jiang-Ling Zhang, Dan Feng, and Zhi-Kun Wan. Cache prefetching adaptive policy based on access pattern. *Proceedings of the First International Conference on Machine Learning and Cybernetics, Beijing*, Vol 1, pp. 496-500, Nov. 2002.
18. Michael Mesnier, Eno Thereska, Daniel Ellard, Gregory R. Ganger, and Margo Seltzer. File Classification in Self-\* Storage Systems. *Proceedings of the First International Conference on Autonomic Computing (ICAC-04)*, New York, May 2004.
19. Craig A.N. Soules, and Greg Ganger. Why Can't I Find My Files? New methods for automating attribute assignment. *Proceedings of the Ninth Workshop on Hot Topics in Operating systems, USENIX Association*, May 2003.
20. P.M. Chen, E.K. Lee, G.A. Gibson, R.H. Katz, and D.A. Patterson, RAID: High Performance and Reliable Secondary Storage, *ACM Computing Surveys*, pp.145-185, 1994.
21. Website, March, 2005, <http://www.ds9a.nl/lvm-howto/>
22. Website, May, 2004, <http://www.snia.org/>
23. T10/1731-D Revision 0, October 2004. SCSI Object-Based Storage Device Commands -2 (OSC-2). Available at <http://www.t10.org>
24. Website, May, 2004, available at <http://sourceforge.net/projects/intel-iscsi/>

# Architecture Based Approach to Adaptable Fault Tolerance in Distributed Object-Oriented Computing

Rodrigo Lanka, Kentaro Oda, and Takaichi Yoshida

Program of Creation Informatics, Kyushu Institute of Technology  
lanka@mickey.ai.kyutech.ac.jp, oda@ci.kyutech.ac.jp,  
takaichi@ai.kyutech.ac.jp

**Abstract.** To gain high level of performance in distributed object oriented computing, a required level of reliability in objects has to be maintained. This brings in a set of complex requirements into consideration. Furthermore depending on the unpredictability of the underlying environment, the replication should have architecture for the adaptable fault tolerance so that it can handle different situations of the underlying system before the system fails. We propose a mechanism for analyzing the complexity of this underlying environments and designing a dynamically reconfigurable architecture. The architecture provides the user required reliability by analyzing the performance and the reliability of the underlying environment and then either adjusting the replication degree or adaptively shifting to a suitable replication protocol. This architecture is a part of the Juice system which supports adaptation properties for a distributed environment.

## 1 Introduction

The system reliability is an important aspect for achieving high performance in distributed computing. Its importance is even further highlighted by the ever increasing usage of distributed systems in many aspects of today's life. However, *maintaining the required reliability in distributed systems* brings in the need for meeting a complex set of requirements. This complexity *depends on the underlying system's reliability* that includes reliabilities of operating system, hardware and network. The underlying system's reliability also fluctuates with *the unpredictable environmental changes*. Some examples of such changes are partial failures of operating systems, hardware failures and network breakdowns.

On the other hand, the required levels of system's reliability may also vary as they are decided by the users where *the cost of the system reliability mainly depends on the environment*, as depicted in Figure 1.

Our main objective is therefore to find both an approach for analyzing the complexities of the underlying environments and a dynamically reconfigurable architecture that supports the adaptive fault tolerance *to provide the required reliability according to changes in the underlying environment*. This would indeed be a significant step forward.

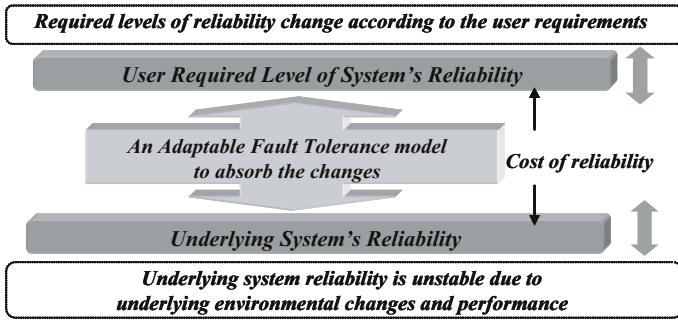


Fig. 1. The adaptable fault tolerance model which absorb the reliability changes

The mechanism that we focus on is *replication* as it is seen as a cost effective way to increase reliability of a system in the presence of potential failures. Moreover, in the distributed computing, replication is used for both performance and fault-tolerant purposes thereby introducing a constant trade-off between efficiency and consistency. However, our argument is that it is still difficult to presume a single replication protocol to get the required reliability throughout the lifetime of an object. Furthermore, maintaining the degree of replication at constant level does not ensure a constant level of reliability as the underlying environment continues to change. Therefore, to absorb the unpredictable changes in the environment, one must consider a suitable architecture for adaptable fault tolerance that is capable of handling different situations of the underlying system. This adaptability would enable the objects to change its respective behaviors that fulfills the current reliability requirements.

The Adaptable Fault Tolerance (AFT) model is based on two main strategies. On the one hand, the system can provide the required reliability by appropriately adjusting the replication degree. Secondly, the reliability of certain systems could be improved by adaptively shifting into a suitable replication protocol. The key factor in strategy selection by AFT model is the trade-off between the cost and the reliability. The AFT model is designed on the Juice object model that comprises adaptability as one of its main features. It allows objects to change its behavior on the fly by replicating some in a modular way [1][2]. The AFT model encapsulates the replication and communication which comprises a group membership service and reliable message delivery service to comply with the consistency of the replication protocol. The AFT model should therefore allow the system to reconfigure itself and maintain the required degree of system reliability. This would enable the system to provide the optimal reliable service even when the underlying environment continues to change.

Most research on reliability estimation assumes that individual component reliabilities are available, which means that they ignore the method of estimating those reliabilities. The reliability estimation models can however be used to project each component's failure rate. But this is not always possible due to the scarcity of failure data [3]. Therefore, our concern is to estimate the underly-

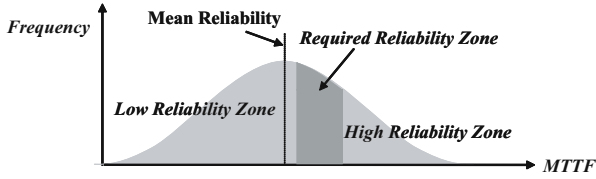


Fig. 2. Underlying System Reliability Zones

ing reliability by taking the entire underlying system as one unit because it has proven to be easy. However, we do not pursue this any further in this paper. For more information refer [3].

Assuming that underlying system reliability can be estimated, we can select strategies by locating it within one of the three reliability zones that are defined as high, required and low (Figure 2).

## 2 The Adaptation Strategies of the Algorithm

It is expected that distributed systems maintain their performance throughout its lifetime, even when it is frequently disturbed by the underlying environmental changes. However, it is impossible to provide a general purpose replication protocol that can be parameterized to accommodate all underlying environmental fluctuations. Therefore, it is important to design an adaptable algorithm to maintain the required reliability for the runtime environment.

The adaptation algorithm can be divided into two main strategies: (a) adjusting the replication degree and (b) changing the replication protocol. When the underlying system falls into the low reliability zone, both strategies can be applied to increase the reliability. But, when the underlying system is in high reliability zone, we need to reduce excess reliability that incurs unnecessary costs. Therefore we reduce the replication degree to remove unwanted reliability.

The reliability of a replicated object is significantly affected by both the number of replicas it has and their placement. Therefore, it appears from the outset that migrating the entire object onto another member who is more reliable than the current member could also be a feasible approach. (The objects are movable as they encapsulate their internal state). However, the object migration itself could increase the complexity of the system therefore we exclude this approach from the AFT model.

### 2.1 Adjust the Replication Degree

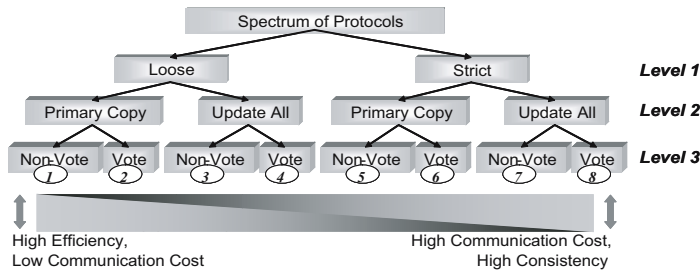
Depending upon the AFT policy, there are two aspects to why an object may change the replication degree.

On the one hand, an object will increase the replication degree when a member is admitted to the group to increase the reliability (i.e. when the reliability of the underlying system goes down; the object creates a new replica member and admits to the group to increase the reliability) or when one recovers from failure.

On the other hand, it will decrease the replication degree when an existing member leaves the group to reduce the reliability, concerned with the additional cost incurred for unwanted reliability or when an existing member fails.

## 2.2 Change the Replication Protocol

**Replication Protocol Classification.** Existing taxonomies of replication techniques take into account a broad spectrum of protocols, including those with both different levels of consistency and reliability. Therefore, the replication protocols can be classified into eight classes based on three parameters (levels) that determine their performance and properties [4]. They are *the method of synchronization*, *the type of protocol* and *the type of communication* (see Figure 3).



**Fig. 3.** Spectrum of replication protocols

*Level 1 - The Synchronization Method (Loose or Strict).* Synchronization depends on both the network traffic generated by the replication algorithm and the overall communication overheads. Therefore, we consider two types of synchronization according to the degree of communication. The *loose synchronization*, synchronizes the replicas by sending a single message containing a set of updates within a given period of time. This is therefore more attractive when dealing with efficiency. In contrast, the *strict synchronization* synchronizes each update individually by sending messages on a per update basis. It is therefore more attractive when dealing with consistency.

*Level 2 - The Protocol Type (Primary Copy or Update-all).* The *Primary copy* replication requires any update to the objects sent to the primary copy first where it is processed. The updates are then propagated to all other back-up sites. Therefore, this is more graceful when dealing with efficiency. In contrast, the *Update-all* replication allows updates to be performed anywhere in the system. That is, updates can concurrently arrive at two different copies of the same data item and it is more graceful when dealing with consistency.

*Level 3 - The Communication Type (Non-voting or Voting).* The method of acknowledging also can be divided into two types: non-voting and voting. Under *non-voting* communication, the object can decide whether to commit or abort a



message by itself. And, this is more graceful, when considering the efficiency. In contrast, *voting* communication requires an extra round of messages to coordinate the different replicas and it helps to provide high consistency.

**Selection of Replication Protocol.** The selection of replication protocols into the runtime environment should be done adaptively from a pool of available protocols depending on the condition of the underlying environment. For selecting the optimal protocol for a particular condition, it is required to estimate the costs of other available protocols and make comparisons. A principle aspect of such comparisons is the trade-off between consistency and cost.

When the AFT model needs shifting the current protocol into a higher efficient one, it shifts to the left from the current position of the spectrum (illustrated by table 1) and vice versa. Thus, one extreme of the above spectrum of replication protocols is represented by those that are highly reliable and efficient but not awfully consistent. And the other extreme is represented by those that guarantee the consistency but prohibitively expensive.

**Table 1.** The shifting methods of the spectrum (*to increase the efficiency*)

Current Protocol	How to Select the Next Protocol		
	<i>Level 1</i>	<i>Level 2</i>	<i>Level 3</i>
8, 6, 4, 2	-	-	Voting $\Rightarrow$ Non-voting
7, 3	-	Update-all $\Rightarrow$ Primary	Non-voting $\Rightarrow$ Voting
5	Strict $\Rightarrow$ Loose	Primary $\Rightarrow$ Update-all	Non-voting $\Rightarrow$ Voting

We assume that all members are connected to become a specified group of replicas and each member carrying a single vote. When the algorithm adopts the majority decision criterion to select the most optimal replication protocol, the members of the replica group take a vote, and if a majority agrees for switching into the new protocol, then they all switch together. The criterion of majority voting requires the agreement from more than one half of the available members. Thus, one member of the replica group, selected at random, gathers the votes and decides the next protocol on behalf of the others.

### 3 The Adaptation Algorithm

The AFT model proposes an adaptation algorithm for executing the switching between strategies, according to the dynamic environmental changes. As an example, suppose when systems reliability goes down, then the algorithm selects one strategy which suits system's reliability requirements. On the other hand, the selection of strategies could be done even according to user requirements (i.e. the user defines the cost and reliability level and the algorithm evaluates and selects the most optimal strategy). As an example, suppose when a user concerns higher level of reliability (or consistency) rather than the cost, or vice versa, then the algorithm can select one strategy which suits user's requirements. The proposed algorithm consists of following steps (Figure 4):

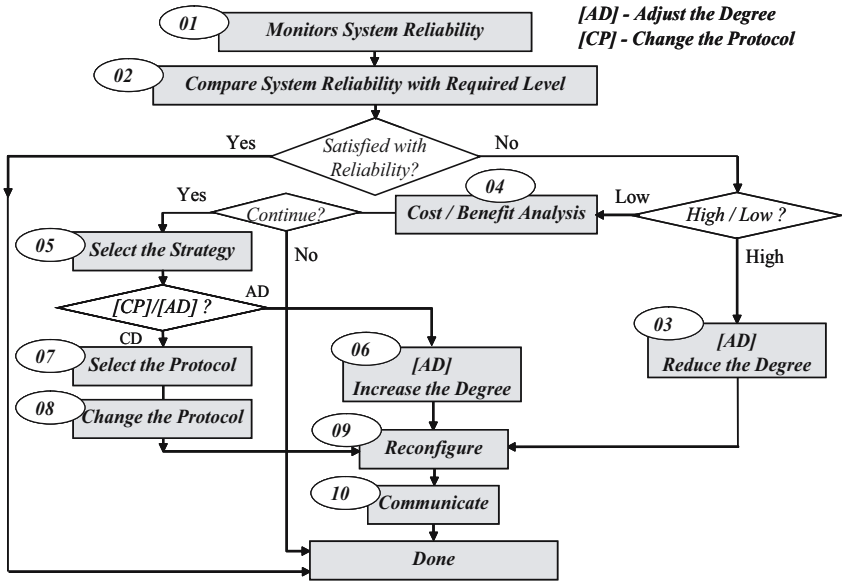


Fig. 4. The adaptable fault tolerance model which absorb the reliability changes

- Step 1.** Each monitor objects (CMI and USI) of the replica members evaluate and monitor the system reliability by using underlying system failure rates, network latency, local CPU load average, system resources and etc.
- Step 2.** Each member compares whether the current system reliability satisfies with either system required or user required reliability level.
- Step 3.** Reduce the current replication protocol degree in order to reduce the unwanted (excess) reliability.
- Step 4.** Carry out a cost-benefit analysis by considering the both user requirements and current system reliability with underlying system properties and its configurations to check whether it is worth to continuing. The impact of the cost of changing strategies and trade-offs need to be carefully studied with different reliability levels.
- Step 5.** Select the optimal strategy between (a) adjusting the degree (AD) and (b) changing the replication protocol (CP) based on their costs. Each member compares the reliability levels and the cost of targeted strategies.
- Step 6.** Increase the current replication protocol degree in order to meet the required level of reliability.
- Step 7.** According to environmental changes and user requirements, select the optimal protocol from the available pool following an agreement amongst the replica group members. For the protocol selection criteria, latency, bandwidth, CPU power, and etc. have to be taken into the consideration.
- Step 8.** Change the protocol when the optimal is selected. One of the members switches the current replication protocol into the optimal protocol.
- Step 9.** Reconfigure the old replication protocol and other configurations (if any) according to the current one.

**Step 10.** The members resume, pending and delivering messages according to the new communication protocol.

However, it is difficult to estimate the total cost of the system, because each member of the replica group is affected by its local environmental changes. Therefore, while each member estimates its local cost by itself, the decision on total cost is made by all. Furthermore, it is difficult to optimize each of the parameters: reliability, adaptability, scalability and consistency individually, which may result in conflicting and inconsistent solutions. For example, the latency might cause messages to arrive in different orders at different machines. Also, the different users getting different views of the underlying environment might result in consistency problems.

## 4 Design on the Juice System

### 4.1 The Adaptable Juice Object Model

The Juice system [1][2] is based on the adaptable object model. Adaptable Juice objects can reconfigure its internal objects with according to the new configurations to adapt with the changing execution environment at runtime. Therefore, this model can support adaptation properties for an open distributed environment. The adaptable object consists of five internal objects (Figure 5).

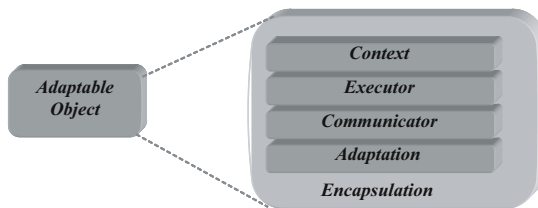
**The Context Object:** It is defined by the application programmer, and deals with the application domain implementation. It holds both the state and behavior of the adaptable object.

**The Executor:** It provides execution and concurrent control. It executes methods from the context object corresponding to the messages received from the communicator.

**The Communicator:** This object interprets the communication protocol. The communicator can be installed modularly to provide a new communication protocol and its semantics.

**The Adaptation Strategy:** It provides strategies for adaptation as environmental changes occur. These changes are informed in the form of events.

**The Encapsulation Object:** It hides the internal structure of the adaptable object. The encapsulation object type is the same as the user-defined type for the problem domain (also same as the type of the context object).



**Fig. 5.** The Adaptable Juice Object Model

### 4.2 The Adaptable Fault Tolerance (AFT) Model

The AFT model needs to provide adaptation for open distributed environments. As this model is designed on the Juice object, it can acquire adaptation as it inherits the properties of the Juice object. Therefore, the components of the AFT model have to be encapsulated within the Juice object to fulfill the requirements of the model. The internal structure of the model is illustrated in Figure 6.

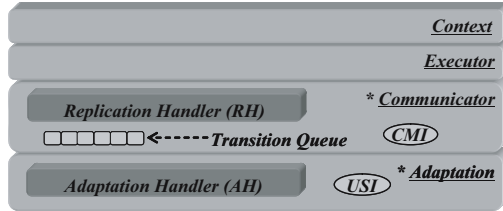


Fig. 6. The Internal Structure of the AFT Model

The model is based on component composition and addresses two levels of configuration: i.e. upper and lower levels. The upper level handles the reconfigurations of the Juice object model components (context, communicator, executor and adaptation) while the lower level reconfigures the objects that lie inside each Juice object model components (replication and adaptation handler).

One of the main advantages of this approach is that, because there are two separate configuration levels, it can help reconfiguring one level of components independently without any involvement of the other level. In other words, it is possible to reconfigure one level without losing the other configurations. As a result of these two reconfigurable structures, this approach helps to perform its switching strategies without leading to any inconsistencies.

In this model, the Underlying System Information (USI) evaluator seeks information from both the underlying virtual machine and operating system (i.e. the information such as available memory, link bandwidth, workload etc.). And, the Client Member Information (CMI) evaluator handles both the most recently connected client’s information and the current replica group members’ information (message failure rate, response time, latency etc.). In other words, both USI and CMI are responsible for obtaining the relevant information and sending to the Adaptation Handler (AH) which resides in the adaptation object of the Juice object model. After receiving the above required information, AH analyses the underlying system reliability and selects the best fit fault tolerance strategy according to the adaptation algorithm, which suites current environmental conditions. This strategy is therefore suitable to provide the required level of reliability in the system. Furthermore, the AH notifies the Replication Handler (RH) to replace themselves with the new object according to the new strategy selected by itself (AH). The RH maintains the replicated Juice object, relevant information about clients/group members and the transition queue.

The RH is in charge of both enforcing the required replication degree and maintaining information about all the clients associated with the replica group. When a strategy is selected for execution, the AH communicates to the RH. The RH then considers the network resources and the number of clients currently in the system, and decides the replication degree needs to be enforced.

### 4.3 Model of Communication

This research further intends to suggest an adaptable client-server group communication model, which can communicate under different environmental conditions with different replication protocols. According to the past researches, there has been a significant progress in the development of group communication infrastructures capable of offering a very impressive range of configuration facilities. For example, (a) BAST [5] allows different protocols to be selected and implemented for the same services under different usage patterns, (b) Horus system [6] allows communication stacks to be changed in runtime and (c) Coyote [7] allows the same message to be processed by different protocols in parallel.

Furthermore, during the transition period no one is responsible for handling messages as they change their configurations. In other words, all the message transactions might be lost during the protocol transition time as communicator of the Juice object reconfigures itself according to the new protocol configurations. Therefore, we designed a mechanism to overcome the problems when replication protocol transition period exists. As a solution all the relevant message transactions will be stored in a message queue called *transition queue* which resides with the communicator object of the Juice model, during the transition period. At this time, the communicator of the Juice model reconfigures itself according to the new strategy selected by AH. And, the transition queue will remain unchanged as it has different levels of configuration. Therefore, after the transition period, all the stored messages can be sent to both the relevant clients and servers with appropriate actions and newly configured values. However, sharing the transition queue between new and old replication handlers (RH) may corrupt the contents of the queue. Therefore, the old RH is not permitted to use the transition queue after it has transferred to the new RH.

## 5 Concluding Remarks

The AFT model with its structure facilitates the maintenance of the required degree of reliability in the system. This enables it to provide the optimal reliable service even when the underlying system changes prevail. This model is designed based on two main strategies: (a) adjusting the replication degree and (b) changing the replication protocol at runtime.

Furthermore, it enables *Juice objects* to reconfigure the current configurations at runtime. The strategies for fault tolerance can be adopted according to the requirements of each adaptable object. The adaptable objects can enhance the reliability of the system whenever changes in the environment turn the working

place to become hostile. It allows the system to reconfigure and execute adaptable fault tolerance algorithm under the current situations of the underlying system. Therefore, unlike common replication protocols, our solution is tightly integrated with the underlying environment.

It also accommodates a replication handler and a transition queue inside the communication object of the Juice system which can handle the problems when the replication protocol transition period exists. The adaptable communication model is also a part of the Juice system.

## References

1. Leonardo, J.C., Oda, K., and Yoshida, T.: An Adaptable Replication Scheme for Reliable Distributed Object-Oriented Computing, 17<sup>th</sup> International Conference on Advanced Information Networking and Applications, (2003)
2. Oda, K., Tazuneki, S., and Yoshida, T.: The Flying Object for an Open Distributed Environment, 15<sup>th</sup> International Conference on Information Networking, (2001)
3. Popstojanova, K.G., and Trivedi, K.S.: Architecture Based Approach to Reliability Assessment of Software Systems, Performance Evaluation, Vol. 45/2-3, (June 2001)
4. Wiesmann, M., Pedone, F., Schiper, A., Kemme, B., and Alonso, G.: Database replication techniques: a three parameter classification, 19<sup>th</sup> IEEE Symposium on Reliable Distributed Systems (SRDS2000), Germany, (October 2000)
5. Garbinato, B., and Guerraoui, R.: Flexible Protocol Composition in Bast, 18<sup>th</sup> International Conference on Distributed Computing Systems (ICDCS-18), (1998)
6. van Renesse, R., Birman, K., Friedman, R., Hayden, M., and Karr, D.: A Framework for Protocol Composition in Horus, Symposium on Principles of Distributed Computing, (1995)
7. Bhatti, N., Hiltunen, M., Schlichting, R., and Chiu, W.: Coyote: A system for constructing fine-grain configurable communication services, ACM Trans. on Computer Systems, 16(4):321-366, (1998)

# Security Analysis of Michael: The IEEE 802.11i Message Integrity Code

Jianyong Huang, Jennifer Seberry, Willy Susilo, and Martin Bunder

University of Wollongong, NSW 2522, Australia  
{jyh33, jennie, wsusilo, mbunder}@uow.edu.au

**Abstract.** The latest IEEE 802.11i uses a keyed hash function, called *Michael*, as the message integrity code. This paper describes some properties and weaknesses of Michael. We provide a necessary and sufficient condition for finding collisions of Michael. Our observation reveals that the collision status of Michael only depends on the second last block message and the output of the block function in the third last round. We show that Michael is not collision-free by providing a method to find collisions of this keyed hash function. Moreover, we develop a method to find fixed points of Michael. If the output of the block function in any round is equal to any of these fixed points, a packet forgery attack could be mounted against Michael. Since the Michael value is encrypted by RC4, the proposed packet forgery attack does not endanger the security of the whole TKIP system.

## 1 Introduction

Wireless devices based on IEEE 802.11b standard [3] are widely in use nowadays. The IEEE 802.11b defines an encryption scheme called Wired Equivalent Privacy (WEP). It is well known that WEP has several serious security flaws. Fluhrer, Mantin, and Shamir [7] (FMS) proposed an attack on the WEP encryption protocol. By exploiting weaknesses of the RC4 [8] key scheduling algorithm, the FMS attack demonstrated that the RC4 encryption key can be easily derived by an eavesdropper who can intercept several million encrypted WEP packets whose first byte of plaintext is known. Stubblefield, Ioannidis, and Rubin [9] practically implemented the FMS attack, and showed that the real systems could be defeated. Borisov, Goldberg, and Wagner [5] showed that the WEP data integrity could be compromised as encrypted messages could be modified by an attacker without being detected. Moreover, Arbaugh, Shankar, and Wan [4] showed that the WEP authentication mechanism is vulnerable to attack.

To address the WEP vulnerabilities, the IEEE 802.11 Task Group i (TGi) provides a short-term solution and a long-term solution. The short-term solution has adopted the Temporal Key Integrity Protocol (TKIP). TKIP is a group of algorithms that wraps the WEP protocol to address the known weaknesses. TKIP includes three components: a message integrity code called *Michael*, a packet sequencing discipline, and a per-packet key mixing function. TKIP is considered as a temporary solution, and it is designed for legacy hardware. For the

long-term solution, the IEEE 802.11 TG1 recommends two modes of operation: WRAP (Wireless Robust Authenticated Protocol) and CCMP (Counter-Mode-CBC-MAC Protocol). Both WRAP and CCMP are based on AES cipher [2], and they require new hardware.

**Our contributions.** In this paper, we investigate the security issues of Michael. First, we present a necessary and sufficient condition for finding collisions of Michael, showing that the collision status of Michael *only* depends on the second last block message and the output of the block function in the third last round. Second, by employing the necessary and sufficient condition, we provide a method to find collisions of Michael and show that Michael is not collision-free. Furthermore, we develop a method to find fixed points of Michael, and a packet forgery attack could be mounted against Michael if the output of the block function in any round is equal to any of these fixed points.

**Notations.** A 64-bit Michael key  $K$  is converted to two 32-bit subkeys,  $k_0$  and  $k_1$ , written as  $K = (k_0, k_1)$ . An  $n$ -block message  $M$  is written as  $M = (m_0, m_1, \dots, m_{n-1})$ .  $L_0^i, R_0^i, L_1^i, R_1^i, L_2^i, R_2^i, L_3^i, R_3^i, L_4^i, R_4^i$ , and  $L_5^i$  are variables used in the  $(i + 1)$ -th round of Michael( $K, M$ ) procedure. For an  $n$ -round Michael( $K, M$ ) procedure, we represent the  $(i + 1)$ -th ( $0 \leq i \leq n - 1$ ) round output of the Michael block function as  $(L_5^i, R_4^i)$ , where  $L_5^i$  stands for the left half of the output and  $R_4^i$  stands for the right half of the output. Some other notations used in this paper are listed as follows:  $\lll$  is left rotation,  $\ggg$  represents right rotation,  $\oplus$  is exclusive-or,  $\boxplus$  stands for addition modulo  $2^{32}$ ,  $\parallel$  is concatenation, and  $\implies$  means “imply”.

**Organization.** The rest of this paper is organized as follows. Section 2 provides the overview of the Michael keyed hash function. Section 3 describes one previous work on Michael, which shows that Michael is invertible. We provide a necessary and sufficient condition for finding collisions of Michael in Section 4. In Section 5, we propose a method to find collisions of Michael, and based on our method, we show that Michael is not collision-free. In Section 6, we introduce a simple method to find fixed points of Michael and propose a packet forgery attack against Michael. Finally, we conclude this paper in Section 7.

## 2 The Michael Keyed Hash Function

Michael [6] is the message integrity code (MIC) of TKIP in the IEEE 802.11i [1]. Michael is a keyed hash function, whose inputs are a 64-bit Michael key and an arbitrarily long message, and output is a 64-bit Michael value. The 64-bit key is converted to two key 32-bit words, and the message is partitioned into 32-bit blocks. The message is padded at the end with a single byte with the hexadecimal value *0x5a* and then followed by between 4 and 7 zero bytes. The number of zero bytes is chosen so that the overall length of the message plus the padding is a multiple of 4. We note that the last block



of the padded message is zero, and the second last block of the padded message is not zero. The details of Michael are described in Algorithm 2.1 and 2.2.

**Algorithm 2.2:**  $B(L, R)$

**Algorithm 2.1:**  $\text{MICHAEL}((k_0, k_1), (m_0, \dots, m_{n-1}))$

**Input :**  $\text{Key}(k_0, k_1)$   
**Input :** **Padded message**  $(m_0, \dots, m_{n-1})$   
**Output :** **MIC value**  $(L, R)$   
 $(L, R) \leftarrow (k_0, k_1)$   
**for**  $i \leftarrow 0$  **to**  $n - 1$   
  **do**  $\begin{cases} L \leftarrow L \oplus m_i \\ (L, R) \leftarrow B(L, R)(\text{Algorithm 2.2}) \end{cases}$   
**return**  $(L, R)$

**Input :**  $(L, R)$   
**Output :**  $(L, R)$   
 $R \leftarrow R \oplus (L \lll 17)$   
 $L \leftarrow (L + R) \bmod 2^{32}$   
 $R \leftarrow R \oplus XSWAP(L)$   
 $L \leftarrow (L + R) \bmod 2^{32}$   
 $R \leftarrow R \oplus (L \lll 3)$   
 $L \leftarrow (L + R) \bmod 2^{32}$   
 $R \leftarrow R \oplus (L \ggg 2)$   
 $L \leftarrow (L + R) \bmod 2^{32}$   
**return**  $(L, R)$

Michael employs several operations, including exclusive-or, left rotation, right rotation, addition modulo  $2^{32}$  and swapping ( $XSWAP$ ). Swapping function  $XSWAP$  swaps the position of the two least significant bytes and the position of the two most significant bytes in a word, i.e.,  $XSWAP(ABCD) = BADC$  where  $A, B, C, D$  are bytes. The block function given in Algorithm 2.2 is an unkeyed 4-round Feistel-type construction.

The TKIP frame appends the MIC value as a tag after the message body. The message body together with the MIC value are encrypted by RC4 at the transmitter and then sent to the receiver. The receiver recomputes the MIC value and compares the computed result with the tag coming with the message. If these two MIC values match, the receiver accepts the message; if not, the receiver rejects the message.

### 3 Related Work

Wool found one weakness of Michael: it is *not* one-way, in fact, it is *invertible* [10]. There exists a simple inverse function, which can recover the secret Michael key  $K$ , given a known message  $M$  and its corresponding Michael value  $\text{MIC} = \text{Michael}(K, M)$ . We note that the block function is unkeyed, and every step in Michael is invertible, therefore the whole Michael algorithm is invertible.

The security of Michael relies on the fact that a message and its hash are encrypted by RC4, and thus the hash value is unknown to the attacker. Wool proposed a related-message attack on Michael [10].

**Remark:** Michael is invertible is known by the inventor of Michael, and this security flaw is mentioned implicitly on Page 14 in [6]: “*a known-plaintext attack will reveal the key stream for that IV, and if the second packet encrypted with the same IV is shorter than the first one, the MIC value is revealed, which can then be used to derive the authentication key.*”

### 4 Finding Collisions of Michael

We study the collision-resistance of Michael in this section. By providing Theorem 1, we prove that the collision status of Michael only depends on the second last block message and the output of the block function in the third last round. We would like to point out that Condition 1 and 2 in Theorem 1 are a necessary and sufficient condition for finding collisions of Michael.

**Theorem 1.** *Given two pairs of keys and messages,  $(Key_1, M_1)$  and  $(Key_2, M_2)$ ,  $Michael(Key_1, M_1) = Michael(Key_2, M_2)$  if and only if the following two conditions hold:*

1.  $R_4^{x-3} = R_4'^{y-3}$
2.  $L_5^{x-3} \oplus L_5'^{y-3} = m_{x-2} \oplus m'_{y-2}$

where  $M_1$  has  $x$  32-bit blocks,  $M_2$  has  $y$  32-bit blocks, and both  $x$  and  $y$  are  $\geq 3$ .

*Proof.* The last three rounds of Michael are illustrated in Figure 1 in Appendix A. We provide the last round and the second last round of  $Michael(Key_1, M_1)$  in Algorithm 4.1 and Algorithm 4.2 respectively. Similarly, the last round and the second last round of  $Michael(Key_2, M_2)$  are shown in Algorithm B.1 and Algorithm B.2 in Appendix B respectively.

**Algorithm 4.1:** LAST ROUND  $(Key_1, M_1)$

1.  $L_0^{x-1} = L_5^{x-2}$
2.  $R_0^{x-1} = R_4^{x-2}$
3.  $L_1^{x-1} = L_0^{x-1} \oplus m_{x-1}$
4.  $R_1^{x-1} = R_0^{x-1} \oplus (L_1^{x-1} \lll 17)$
5.  $L_2^{x-1} = (L_1^{x-1} + R_1^{x-1}) \bmod 2^{32}$
6.  $R_2^{x-1} = R_1^{x-1} \oplus XSWAP(L_2^{x-1})$
7.  $L_3^{x-1} = (L_2^{x-1} + R_2^{x-1}) \bmod 2^{32}$
8.  $R_3^{x-1} = R_2^{x-1} \oplus (L_3^{x-1} \lll 3)$
9.  $L_4^{x-1} = (L_3^{x-1} + R_3^{x-1}) \bmod 2^{32}$
10.  $R_4^{x-1} = R_3^{x-1} \oplus (L_4^{x-1} \ggg 2)$
11.  $L_5^{x-1} = (L_4^{x-1} + R_4^{x-1}) \bmod 2^{32}$   
 (Note :  $Michael(Key_1, M_1) = (L_5^{x-1}, R_4^{x-1})$ )

**Algorithm 4.2:** 2RD LAST  $(Key_1, M_1)$

1.  $L_0^{x-2} = L_5^{x-3}$
2.  $R_0^{x-2} = R_4^{x-3}$
3.  $L_1^{x-2} = L_0^{x-2} \oplus m_{x-2}$
4.  $R_1^{x-2} = R_0^{x-2} \oplus (L_1^{x-2} \lll 17)$
5.  $L_2^{x-2} = (L_1^{x-2} + R_1^{x-2}) \bmod 2^{32}$
6.  $R_2^{x-2} = R_1^{x-2} \oplus XSWAP(L_2^{x-2})$
7.  $L_3^{x-2} = (L_2^{x-2} + R_2^{x-2}) \bmod 2^{32}$
8.  $R_3^{x-2} = R_2^{x-2} \oplus (L_3^{x-2} \lll 3)$
9.  $L_4^{x-2} = (L_3^{x-2} + R_3^{x-2}) \bmod 2^{32}$
10.  $R_4^{x-2} = R_3^{x-2} \oplus (L_4^{x-2} \ggg 2)$
11.  $L_5^{x-2} = (L_4^{x-2} + R_4^{x-2}) \bmod 2^{32}$

**Necessary condition:** If  $Michael(Key_1, M_1) = Michael(Key_2, M_2)$ , namely the collisions occur, we then backtrack from Step 11 and 10 in Algorithm 4.1 and B.1.

$$\begin{aligned}
 L_5^{x-1} &= L_5^{y-1} \text{ and } R_4^{x-1} = R_4^{y-1} \implies L_4^{x-1} = L_4^{y-1}, \\
 L_4^{x-1} &= L_4^{y-1} \text{ and } R_4^{x-1} = R_4^{y-1} \implies R_3^{x-1} = R_3^{y-1}, \\
 L_4^{x-1} &= L_4^{y-1} \text{ and } R_3^{x-1} = R_3^{y-1} \implies L_3^{x-1} = L_3^{y-1}, \\
 L_3^{x-1} &= L_3^{y-1} \text{ and } R_3^{x-1} = R_3^{y-1} \implies R_2^{x-1} = R_2^{y-1}, \\
 L_3^{x-1} &= L_3^{y-1} \text{ and } R_2^{x-1} = R_2^{y-1} \implies L_2^{x-1} = L_2^{y-1}, \\
 L_2^{x-1} &= L_2^{y-1} \text{ and } R_2^{x-1} = R_2^{y-1} \implies R_1^{x-1} = R_1^{y-1},
 \end{aligned}$$

$$\begin{aligned} L_2^{x-1} = L_2^{y-1} \text{ and } R_1^{x-1} = R_1^{y-1} &\implies L_1^{x-1} = L_1^{y-1}, \\ L_1^{x-1} = L_1^{y-1} \text{ and } R_1^{x-1} = R_1^{y-1} &\implies R_0^{x-1} = R_0^{y-1}. \end{aligned}$$

As  $L_1^{x-1} = L_0^{x-1} \oplus m_{x-1}$ ,  $L_1^{y-1} = L_0^{y-1} \oplus m'_{y-1}$ ,  $L_0^{x-1} = L_5^{x-2}$ ,  $L_0^{y-1} = L_5^{y-2}$ ,  $R_0^{x-1} = R_4^{x-2}$ ,  $R_0^{y-1} = R_4^{y-2}$ ,  $m_{x-1} = 0$  and  $m'_{y-1} = 0$ , therefore  $L_5^{x-2} = L_5^{y-2}$  and  $R_4^{x-2} = R_4^{y-2}$ .

Similarly, we use the same method in the second last rounds of Michael( $Key_1$ ,  $M_1$ ) and Michael( $Key_2$ ,  $M_2$ ).

$$\begin{aligned} L_5^{x-2} = L_5^{y-2} \text{ and } R_4^{x-2} = R_4^{y-2} &\implies L_4^{x-2} = L_4^{y-2}, \\ L_4^{x-2} = L_4^{y-2} \text{ and } R_4^{x-2} = R_4^{y-2} &\implies R_3^{x-2} = R_3^{y-2}, \\ L_4^{x-2} = L_4^{y-2} \text{ and } R_3^{x-2} = R_3^{y-2} &\implies L_3^{x-2} = L_3^{y-2}, \\ L_3^{x-2} = L_3^{y-2} \text{ and } R_3^{x-2} = R_3^{y-2} &\implies R_2^{x-2} = R_2^{y-2}, \\ L_3^{x-2} = L_3^{y-2} \text{ and } R_2^{x-2} = R_2^{y-2} &\implies L_2^{x-2} = L_2^{y-2}, \\ L_2^{x-2} = L_2^{y-2} \text{ and } R_2^{x-2} = R_2^{y-2} &\implies R_1^{x-2} = R_1^{y-2}, \\ L_2^{x-2} = L_2^{y-2} \text{ and } R_1^{x-2} = R_1^{y-2} &\implies L_1^{x-2} = L_1^{y-2}, \\ L_1^{x-2} = L_1^{y-2} \text{ and } R_1^{x-2} = R_1^{y-2} &\implies R_0^{x-2} = R_0^{y-2}. \end{aligned}$$

As  $L_1^{x-2} = L_0^{x-2} \oplus m_{x-2}$ ,  $L_1^{y-2} = L_0^{y-2} \oplus m'_{y-2}$ ,  $L_0^{x-2} = L_5^{x-3}$  and  $L_0^{y-2} = L_5^{y-3}$ , therefore  $L_5^{x-3} \oplus L_5^{y-3} = m_{x-2} \oplus m'_{y-2}$ . As  $R_0^{x-2} = R_4^{x-3}$  and  $R_0^{y-2} = R_4^{y-3}$ , therefore  $R_4^{x-3} = R_4^{y-3}$ .

Thus, Michael( $Key_1$ ,  $M_1$ ) = Michael( $Key_2$ ,  $M_2$ )  $\implies R_4^{x-3} = R_4^{y-3}$  and  $L_5^{x-3} \oplus L_5^{y-3} = m_{x-2} \oplus m'_{y-2}$ .

**Sufficient condition:** If  $R_4^{x-3} = R_4^{y-3}$  and  $L_5^{x-3} \oplus L_5^{y-3} = m_{x-2} \oplus m'_{y-2}$  hold, we start from Step 1 and 2 in Algorithm 4.2 and B.2.

$$\begin{aligned} L_5^{x-3} = L_0^{x-2}, L_5^{y-3} = L_0^{y-2} \text{ and } L_5^{x-3} \oplus L_5^{y-3} = m_{x-2} \oplus m'_{y-2} &\implies \\ L_1^{x-2} = L_1^{y-2}, & \\ R_4^{x-3} = R_4^{y-3}, R_4^{x-3} = R_0^{x-2} \text{ and } R_4^{y-3} = R_0^{y-2} &\implies R_0^{x-2} = R_0^{y-2}, \\ L_1^{x-2} = L_1^{y-2} \text{ and } R_0^{x-2} = R_0^{y-2} &\implies R_1^{x-2} = R_1^{y-2}, \\ L_1^{x-2} = L_1^{y-2} \text{ and } R_1^{x-2} = R_1^{y-2} &\implies L_2^{x-2} = L_2^{y-2}, \\ L_2^{x-2} = L_2^{y-2} \text{ and } R_1^{x-2} = R_1^{y-2} &\implies R_2^{x-2} = R_2^{y-2}, \\ L_2^{x-2} = L_2^{y-2} \text{ and } R_2^{x-2} = R_2^{y-2} &\implies L_3^{x-2} = L_3^{y-2}, \\ L_3^{x-2} = L_3^{y-2} \text{ and } R_2^{x-2} = R_2^{y-2} &\implies R_3^{x-2} = R_3^{y-2}, \\ L_3^{x-2} = L_3^{y-2} \text{ and } R_3^{x-2} = R_3^{y-2} &\implies L_4^{x-2} = L_4^{y-2}, \\ L_4^{x-2} = L_4^{y-2} \text{ and } R_3^{x-2} = R_3^{y-2} &\implies R_4^{x-2} = R_4^{y-2}, \\ L_4^{x-2} = L_4^{y-2} \text{ and } R_4^{x-2} = R_4^{y-2} &\implies L_5^{x-2} = L_5^{y-2}. \end{aligned}$$

Finally, we bring the above results from the second last rounds to the last rounds. According to the padding method, we note that  $m_{x-1} = 0$  and  $m'_{y-1} = 0$ .

$$\begin{aligned} L_5^{x-2} = L_5^{y-2}, L_0^{x-1} = L_5^{x-2} \text{ and } L_0^{y-1} = L_5^{y-2} &\implies L_0^{x-1} = L_0^{y-1}, \\ R_4^{x-2} = R_4^{y-2}, R_4^{x-2} = R_0^{x-1} \text{ and } R_4^{y-2} = R_0^{y-1} &\implies R_0^{x-1} = R_0^{y-1}, \\ L_0^{x-1} = L_0^{y-1} \text{ and } m_{x-1} = m'_{y-1} &\implies L_1^{x-1} = L_1^{y-1}, \\ L_1^{x-1} = L_1^{y-1} \text{ and } R_0^{x-1} = R_0^{y-1} &\implies R_1^{x-1} = R_1^{y-1}, \\ L_1^{x-1} = L_1^{y-1} \text{ and } R_1^{x-1} = R_1^{y-1} &\implies L_2^{x-1} = L_2^{y-1}, \end{aligned}$$

$$\begin{aligned}
 L_2^{x-1} &= L_2^{y-1} \text{ and } R_1^{x-1} = R_1^{y-1} \implies R_2^{x-1} = R_2^{y-1}, \\
 L_2^{x-1} &= L_2^{y-1} \text{ and } R_2^{x-1} = R_2^{y-1} \implies L_3^{x-1} = L_3^{y-1}, \\
 L_3^{x-1} &= L_3^{y-1} \text{ and } R_2^{x-1} = R_2^{y-1} \implies R_3^{x-1} = R_3^{y-1}, \\
 L_3^{x-1} &= L_3^{y-1} \text{ and } R_3^{x-1} = R_3^{y-1} \implies L_4^{x-1} = L_4^{y-1}, \\
 L_4^{x-1} &= L_4^{y-1} \text{ and } R_3^{x-1} = R_3^{y-1} \implies R_4^{x-1} = R_4^{y-1}, \\
 L_4^{x-1} &= L_4^{y-1} \text{ and } R_4^{x-1} = R_4^{y-1} \implies L_5^{x-1} = L_5^{y-1}.
 \end{aligned}$$

Therefore,  $R_4^{x-3} = R_4^{y-3}$  and  $L_5^{x-3} \oplus L_5^{y-3} = m_{x-2} \oplus m'_{y-2} \implies \text{Michael}(Key_1, M_1) = \text{Michael}(Key_2, M_2)$ .

Therefore,  $R_4^{x-3} = R_4^{y-3}$  and  $L_5^{x-3} \oplus L_5^{y-3} = m_{x-2} \oplus m'_{y-2}$  are a necessary and sufficient condition of  $\text{Michael}(Key_1, M_1) = \text{Michael}(Key_2, M_2)$ .  $\square$

### 5 Michael Is Not Collision-Free

In this section, we show that Michael is not collision-free by providing a simple method to find collisions of Michael. Intuitively, for a given arbitrarily length message  $M$  and a key  $K$ , a 96-bit block message  $M'$  and a key  $K'$  can be computed such that  $\text{Michael}(K, M) = \text{Michael}(K', M')$ .

**Theorem 2.** *Given an arbitrarily length message  $M$  and a specific key  $K$ , a 96-bit block message  $M'$  distinct from  $M$  and a key  $K'$  can always be computed such that  $\text{Michael}(K, M) = \text{Michael}(K', M')$ , where  $M$  has  $n$  32-bit blocks and  $n$  is any integer  $\geq 3$ .*

*Proof.* We write  $M$  as  $(m_0, m_1, \dots, m_{n-1})$ , and  $M'$  as  $(m'_0, m'_1, m'_2)$ . We represent the outputs of the last, second last, third last and fourth last round of  $\text{Michael}(K, M)$  as  $(L_5^{n-1}, R_4^{n-1})$ ,  $(L_5^{n-2}, R_4^{n-2})$ ,  $(L_5^{n-3}, R_4^{n-3})$  and  $(L_5^{n-4}, R_4^{n-4})$  respectively. The outputs of the last, second last and third last round of  $\text{Michael}(K', M')$  are represented as  $(L_5^2, R_4^2)$ ,  $(L_5^1, R_4^1)$  and  $(L_5^0, R_4^0)$  respectively.  $K'$  is written as  $(k'_0, k'_1)$ .  $K'$ ,  $m'_0$ ,  $m'_1$  and  $m'_2$  are constructed as follows.

1. Choose  $m'_2 = 0$  (as  $m_{n-1} = 0$  according to the padding method).
2. Choose  $m'_1 = m_{n-2}$ .
3. Choose  $m'_0$  arbitrarily, but  $m'_0 \neq m_{n-3}$  if  $n = 3$ .
4. Choose  $k'_0 = L_5^{n-4} \oplus m_{n-3} \oplus m'_0$  and  $k'_1 = R_4^{n-4}$ .  $K'$  is constructed as  $K' = (k'_0, k'_1) = (L_5^{n-4} \oplus m_{n-3} \oplus m'_0, R_4^{n-4})$ .

The construction is illustrated in Figure 2 in Appendix A. The soundness of this construction is shown as follows.

$$\begin{aligned}
 k'_0 &= L_5^{n-4} \oplus m_{n-3} \oplus m'_0 \implies k'_0 \oplus m'_0 = L_5^{n-4} \oplus m_{n-3}, \\
 k'_0 \oplus m'_0 &= L_5^{n-4} \oplus m_{n-3} \text{ and } k'_1 = R_4^{n-4} \implies R_4^{n-3} = R_4^0 \text{ and } L_5^{n-3} = L_5^0, \\
 L_5^{n-3} &= L_5^0 \text{ and } m_{n-2} = m'_1 \implies L_5^{n-3} \oplus L_5^0 = m_{n-2} \oplus m'_1.
 \end{aligned}$$

Therefore,  $\text{Michael}(K, M) = \text{Michael}(K', M')$  holds because  $R_4^{n-3} = R_4^0$  satisfies Condition 1 in Theorem 1 and  $L_5^{n-3} \oplus L_5^0 = m_{n-2} \oplus m'_1$  satisfies Condition 2 in Theorem 1.  $\square$

**Theorem 3.** *Michael is not collision-free.*

*Proof.* Can be deduced from Theorem 2.  $\square$

## 6 Finding Fixed Points of Michael

In this section, we present a method to find fixed points of Michael. A fixed point of Michael is a triple  $(L_i, R_i, m_i)$  such that  $\text{Michael}((L_i, R_i), m_i) = (L_i, R_i)$ . The procedure is described in Section 6.1. A packet forgery attack could be mounted against Michael if the output of the Michael block function is equal to any of the fixed points. The packet forgery attack is shown in Section 6.2.

### 6.1 The Fixed-Point Finding Procedure

To find fixed points of Michael, we only need to focus on one round of Michael. Figure 3 in Appendix C illustrates one round of Michael. In Figure 3, we note that  $\text{Michael}((L_i, R_i), m_i) = (L_{i+1}, R_{i+1})$ . In the finding procedure, our goal is to find a triple  $(L_i, R_i, m_i)$  such that  $\text{Michael}((L_i, R_i), m_i) = (L_{i+1}, R_{i+1}) = (L_i, R_i)$ . The procedure is described as follows.

1. Let  $X_i = L_i \oplus m_i$ , and choose a value for  $R_i$ . Define a counter  $c$  and set it to zero.
2. FOR ( $X_i = 0$ ;  $X_i \leq 2^{32}$ ;  $X_i++$ )
  - (a) Call block function  $B(X_i, R_i)$
  - (b) IF  $R_i = R_{i+1}$  THEN
    - i. There exists an  $X_i$  such that  $R_i = R_{i+1}$ . For a found  $X_i$ , there exists a corresponding  $L_{i+1}$  because the mapping from  $(X_i, R_i)$  to  $(L_{i+1}, R_{i+1})$  is bijective. Choose  $L_i = L_{i+1}$ .
    - ii. Choose  $m_i = X_i \oplus L_i$ .
    - iii. Increase counter  $c$  by one.
3. IF counter  $c = 0$  THEN no fixed point found for this  $R_i$ .
4. ELSE There are  $c$  fixed points for this  $R_i$ .

The key point of this procedure is in Step 2 (b). Given an  $X_i$ , if  $R_i = R_{i+1}$  holds, there exists a fixed point  $(m_i, L_i, R_i)$  such that  $\text{Michael}((L_i, R_i), m_i) = (L_i, R_i)$ . For a specific value of  $R_i$ , the time complexity of deciding whether there exists a fixed point of Michael is  $O(2^{32})$ . To search the complete space of  $R_i$  for all fixed points, the time complexity is  $O(2^{64})$  since  $R_i$  is 32-bit.

We have implemented the fixed-point finding procedure on a personal computer whose processor is an Intel Pentium 4 2.8 GHz, and the program takes 2-3 minutes to decide whether there exists a fixed point for a given  $R_i$ . For example,  $(L_i, R_i, m_i) = (0x3f651087, 0x2, 0xbbac8b1a)$  is a fixed point. A more complete fixed-point table is provided in the full paper.

### 6.2 A Packet Forgery Attack

A packet forgery attack (depicted in Figure 4 in Appendix C) could be mounted against Michael if the output of the block function in any round is equal to any of the fixed points.

**Theorem 4.** *Given a message  $M_1$  and an arbitrary key  $K$ , an attacker can always construct a message  $M_2$  distinct from  $M_1$  such that  $\text{Michael}(K, M_1) = \text{Michael}(K, M_2)$  if the following condition holds.*

1. The output of the block function of Michael( $K, M_1$ ) in any round is equal to any of the fixed points.

*Proof.* Suppose  $M_1$  has  $n$  blocks, and is written as  $(m_0, m_1, \dots, m_{n-1})$ . Suppose the output of block function in any round, say in the  $(i + 1)$ -th round (the corresponding message is  $m_i$ ), is equal to any of the fixed points (assume this point is  $(L_i, R_i)$ ). Given a fixed point  $(L_i, R_i)$ , we can find a corresponding  $m'_i$  from the fixed-point table. A multiple of four blocks of message  $m'_i$  can be appended to the  $(i+1)$ -th round without changing the Michael value. The reason why the number of the inserted blocks of  $m'_i$  is a multiple of four is due to the padding method of Michael. In other words, we need to guarantee  $\text{length}(M_1) \bmod 4 = \text{length}(M_2) \bmod 4$ . Thus,  $M_2$  can be constructed as  $(m_0, m_1, \dots, m_i, < m'_i, m'_i, \dots, m'_i, >, m_{i+1}, \dots, m_{n-1})$ , where the number of the inserted blocks of  $m'_i$  is a multiple of four. According to the property of fixed points, we have  $\text{Michael}(K, M_1) = \text{Michael}(K, M_2)$ .  $\square$

**Remark:** 1. If Condition 1 in Theorem 4 holds, an attacker can forge a message  $M_2$  to replace the original message  $M_1$  without modifying the Michael value, and this packet forgery attack can apply to any key  $K$ . 2. We note that the packet forgery attack does not endanger the entire TKIP system as the message and the hash value are encrypted by RC4. Hence an attacker needs to know the decryption before mounting such a forgery attack against Michael.

## 7 Conclusions

Michael was designed as the message integrity code for the IEEE 802.11i. In this paper, by providing a necessary and sufficient condition for finding collisions of Michael, we showed that the collision status of Michael only depends on the second last block message and the output of its third last round. Therefore, to find collisions of Michael, we only need to focus on its two rounds: the third last round and the second last round. In addition, we demonstrated that Michael is not collision-free. Moreover, we proposed a simple method to find fixed points of Michael and built a fixed-point table based on our results. If the output of the block function in any round is in the fixed-point table, a packet forgery attack could be mounted against Michael. The packet forgery attack does not endanger security of the whole TKIP system as the Michael value is encrypted by RC4. To make the proposed forgery attack practical to TKIP, the attacker needs to consider the combination of Michael and RC4.

## References

1. Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Medium Access Control (MAC) Security Enhancements. 23 July 2004.

2. Advanced Encryption Standard. National Institute of Standards and Technology, NIST FIPS PUB 197, U.S. Department of Commerce. November 2001.
3. ANSI/IEEE Std 802.11, 1999 Edition. Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.
4. W. Arbaugh, N. Shankar, and Y.C. Wan. Your 802.11 Wireless Network has No Clothes. In *Proceedings of IEEE International Conference on Wireless LANs and Home Networks*, pages 131–144, Singapore, 2001.
5. N. Borisov, I. Goldberg, and D. Wagner. Intercepting Mobile Communications: The Insecurity of 802.11. In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, pages 180–189, Rome, Italy, 2001.
6. N. Ferguson. Michael: an improved MIC for 802.11 WEP. *IEEE 802.11 doc 02-020r0*, 17 January 2002. <http://grouper.ieee.org/groups/802/11/Documents/DocumentHolder/2-020.zip>.
7. S. Fluhrer, I. Mantin, and A. Shamir. Weaknesses in the Key Scheduling Algorithm of RC4. In *Proceedings of the 8th Annual International Workshop on Selected Areas in Cryptography*, pages 1–24, Toronto, Canada, 2001.
8. R. Rivest. The RC4 Encryption Algorithm, RSA Data Security Inc., (Proprietary). March 1992.
9. A. Stubblefield, J. Ioannidis, and A. Rubin. Using the Fluhrer, Mantin, and Shamir Attack to Break WEP. In *Proceedings of the 2002 Network and Distributed Systems Security Symposium*, pages 17–22, San Diego, California, 2002.
10. A. Wool. A Note on the Fragility of the “Michael” Message Integrity Code. *IEEE Transactions on Wireless Communications*, 2004.

## A Three-Round Diagrams

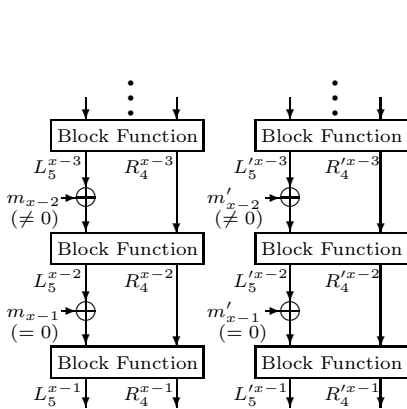


Fig. 1. Last Three Rounds of Michael

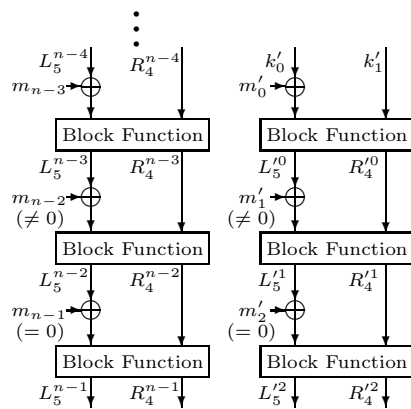


Fig. 2. The Construction of  $(K', M')$

## B Algorithms in Section 4

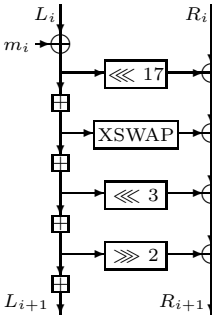
**Algorithm B.1:** LAST ROUND ( $Key_2, M_2$ )

1.  $L_0^{y-1} = L_5^{y-2}$
  2.  $R_0^{y-1} = R_4^{y-2}$
  3.  $L_1^{y-1} = L_0^{y-1} \oplus m'_{y-1}$
  4.  $R_1^{y-1} = R_0^{y-1} \oplus (L_1^{y-1} \lll 17)$
  5.  $L_2^{y-1} = (L_1^{y-1} + R_1^{y-1}) \bmod 2^{32}$
  6.  $R_2^{y-1} = R_1^{y-1} \oplus XSWAP(L_2^{y-1})$
  7.  $L_3^{y-1} = (L_2^{y-1} + R_2^{y-1}) \bmod 2^{32}$
  8.  $R_3^{y-1} = R_2^{y-1} \oplus (L_3^{y-1} \lll 3)$
  9.  $L_4^{y-1} = (L_3^{y-1} + R_3^{y-1}) \bmod 2^{32}$
  10.  $R_4^{y-1} = R_3^{y-1} \oplus (L_4^{y-1} \ggg 2)$
  11.  $L_5^{y-1} = (L_4^{y-1} + R_4^{y-1}) \bmod 2^{32}$
- (Note :  $Michael(Key_2, M_2) = (L_5^{y-1}, R_4^{y-1})$ )

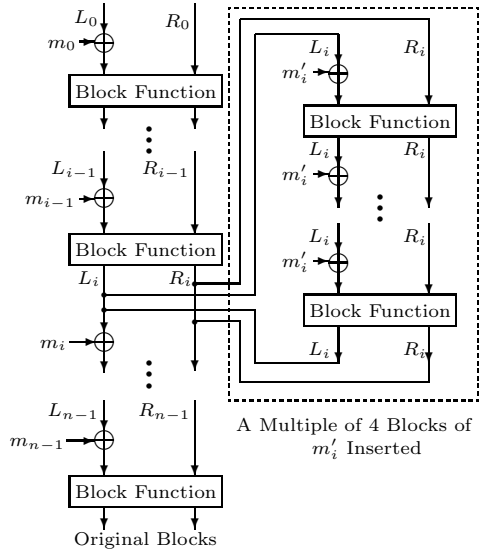
**Algorithm B.2:** 2RD LAST ( $Key_2, M_2$ )

1.  $L_0^{y-2} = L_5^{y-3}$
2.  $R_0^{y-2} = R_4^{y-3}$
3.  $L_1^{y-2} = L_0^{y-2} \oplus m'_{y-2}$
4.  $R_1^{y-2} = R_0^{y-2} \oplus (L_1^{y-2} \lll 17)$
5.  $L_2^{y-2} = (L_1^{y-2} + R_1^{y-2}) \bmod 2^{32}$
6.  $R_2^{y-2} = R_1^{y-2} \oplus XSWAP(L_2^{y-2})$
7.  $L_3^{y-2} = (L_2^{y-2} + R_2^{y-2}) \bmod 2^{32}$
8.  $R_3^{y-2} = R_2^{y-2} \oplus (L_3^{y-2} \lll 3)$
9.  $L_4^{y-2} = (L_3^{y-2} + R_3^{y-2}) \bmod 2^{32}$
10.  $R_4^{y-2} = R_3^{y-2} \oplus (L_4^{y-2} \ggg 2)$
11.  $L_5^{y-2} = (L_4^{y-2} + R_4^{y-2}) \bmod 2^{32}$

## C Figures



**Fig. 3.** One Round of Michael



**Fig. 4.** The Packet Forgery Attack



# A Framework for Protecting Private Information Through User-Trusted-Program and Its Realizability

Ken'ichi Takahashi<sup>1</sup>, Kouichi Sakurai<sup>1,2</sup>, and Makoto Amamiya<sup>2</sup>

<sup>1</sup> Institute of Systems & Information Technologies/KYUSHU,  
2-1-22 Momochihama, Sawara-ku, Fukuoka 814-0001, Japan  
{takahashi, sakurai}@isit.or.jp

<sup>2</sup> Faculty of Information Science and Electrical Engineering, Kyushu University,  
6-1 Kasuga-Koen, Kasuga-shi, Fukuoka 816-8580, Japan  
amamiya@al.is.kyushu-u.ac.jp

**Abstract.** Thanks to the spread of mobile technologies, we will be able to realize the ubiquitous computing environment, in which equipment connected to the Internet assists users in their activities without special care on their part. Then, a function to protect private information is needed. This paper proposes a model for protecting private information. The basic idea of our model is to make use of private information through a program which a user trusts. A user offers a trusted program to a partner and compels a partner to make use of his private information through this program. In this way, a user prevents illegal use of his private information.

## 1 Introduction

Thanks to the spread of mobile technologies, people can access the Internet anytime through their cellular phones. In the near future, many equipments, for example cellular phones, refrigerators, microwave ovens, etc, will be connected to the Internet. This will enable the realization of the ubiquitous computing environment, in which equipment connected to the Internet assists users in their activities.

In the ubiquitous computing environment, there are a good many services which assist users. Then, it is difficult to find services which a user wants to use from among such a vast number of services. There are services which are universally available, and also are provided only to specific users. There are services which a user wants to use and also services which the user never use. Therefore, it is necessary to show services which the user can use and wants to use. Then, it is necessary to compare the substance of services with a user's private information. For example, the service in a liquor shop should be provided only for adults, not provided for children. Also, users who do not drink liquor will never use the service. Thus, the service should be offered only to users who are adults and drink liquor. Therefore, the user must reveal his age to the

liquor shop for approval. Moreover it needs to judge whether he drinks liquor or not. But since this is private information, he may not want to reveal it. Also, some services may require users' private information. For example, online sales will require buyers' credit card information for payments. But credit card information is most important private information, and a user never wants it to be used for purposes other than the payment of his purchases. Therefore users must protect private information from illegal use by service providers.

As mentioned above, we need to provide private information to a service provider for service use and for the selection of services, and then we must protect private information. In this paper, we propose a model for protecting private information. The basic idea of our model is to make available private information through a program which a user trusts. A user offers his trusted program to a service provider and compels the service provider to make use of his private information through that program. In this way, a user will be able to prevent illegal uses of his private information by a service provider.

## 2 Related Work

Cryptographic algorithms, such as symmetric algorithms and public-key algorithms, and technologies based on them, such as digital signatures and Public Key Infrastructure (PKI), have been proposed. These algorithms and technologies aim at the prevention of message interception or the identification of communication partners. Therefore we can ensure message confidentiality, integrity and availability against malicious exploitation by third parties. But they are difficult to prevent illegal uses of released information by a communication partner.

At the bottom of the homepages, we often find links concerning privacy, shown as *Privacy Policy* at Yahoo!, *Privacy* at IBM and so on. These pages show how the company treats users' information that the company collects. Here, there are two problems. One is that users must read the privacy page carefully. Most people will not read the page, even when they provide their information. Another one is that we cannot confirm whether the company actually keeps the promises made on the privacy page or not. Consequently, we have no choice but to believe that the company will keep the promises written on the privacy page.

The Platform for Private Preferences (P3P) [4] enables web sites to express their privacy policies in a standard format that can be interpreted automatically by user agents. Thus, a user agent can automate decision-making by the comparison between a privacy policy and user-specified privacy preferences. So that, users do not need to read the privacy policies at every site they visit. P3P, however, does not provide technical assurance that sites act according to their privacy policies.

The Enterprise Privacy Authorization Language (EPAL) [7] is a formal language to specify fine-grained enterprise privacy policies. An EPAL policy defines format privacy authorization rules that allow or deny actions on data-categories by user-categories for certain purposes under certain conditions while mandating certain obligations. Employees within the organization are compelled to keep

EPAL policies. Thus EPAL prevents the illegal use of information by employees within the organization. But EPAL does not enable users to consent to how the organization protects their private information. Consequently, users cannot know whether the organization manages private information securely.

### 3 A Basic Model for Protecting Private Information

A service may need to check user's private information. Therefore, a user has to reveal some private information, such as a credit card number, his name and so on, if he wishes to use the service. Then, we must protect private information which we released.

#### 3.1 Ways of Checking Private Information

The best way to protect private information is not to release private information to others (Fig. 1.a). Then private information will be checked by the user self. Therefore, a user does not need to worry about the risks of releasing private information. But we can apply this method only in situations where service providers do not need exact verification of users' private information. In other words, this method is not appropriate in situations that need a password check or a check of a right to use the service.

The next method is to release private information only to a trusted third party [3] (Fig. 1.b). The service provider commits the verification of private information to a trusted third party. And the user releases his private information to the trusted third party. Consequently, a service provider gets the result of the verification from the trusted third party without actually getting the private information. Then the user does not need to worry about the risks of releasing private information. But we must provide a third party that both of them can trust. It is difficult to provide a third party that any pair of users and service providers can trust. Even if we prepares some third parties which both of them can trust, we must worry that responsibility and the computational load are centralized in their third parties.

The last method is to release private information to a service provider (Fig. 1.c). If a user can trust a service provider, he may not need to worry about the risks of releasing private information. However, even if we assume a hierarchical PKI model, it is difficult to construct a mechanism that enable any user to trust any service provider. Therefore, we should also assume that a user cannot trust a service provider. If private information is released once to the other, the user cannot manage it. For this reason, it is risky to release private information to such an untrusted service provider, even if a user wants to use the service. Hence, we need a method that enables a service provider to verify private information (e.g. passwords and a right to use the service) and enables a user to prevent illegal use of private information.

To satisfy this dual requirement, we propose a method in which a user requires a service provider to make use of his private information through programs which

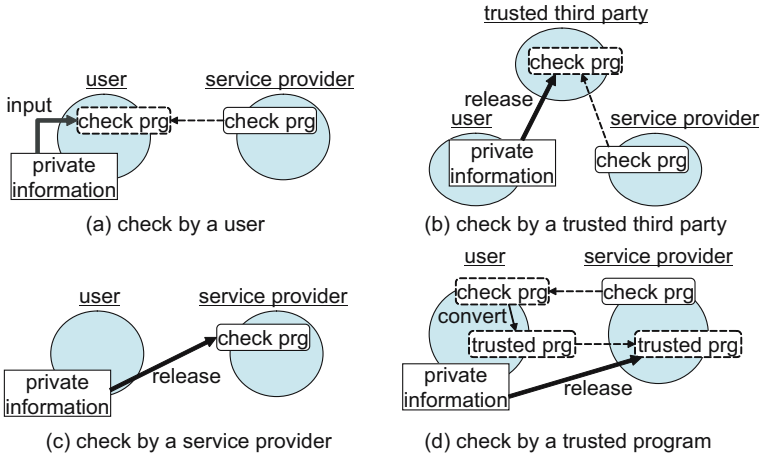


Fig. 1. Models for protecting Private Information

he can trust (Fig. 1.d). A user confirms that a program does not leak information, furthermore, is not used for purposes other than his wishes. Then, he requires the service provider to make use of his private information through the program. In this way, a user will be able to prevent illegal use of private information by the service provider.

### 3.2 The Public and Private Zone Model

We introduce the *public zone* and the *private zone* model [6] as a model for protecting private information based on verification by a user-trusted program. Our model is based on two agent systems, named KODAMA [8] and VPC [2]. In our model, users and service providers are represented as agents. An agent has a public zone and a private zone. The public zone is a freely accessible space and realizes flexible service use. The private zone manages and protects private information. And a *security barrier* exists between the public zone and the private zone. The security barrier has functions for restricting the access to private information and for the control of communications of a program which accesses private information. An overview of our model is illustrated in Fig. 2.

**The Public Zone:** In the ubiquitous computing environment, there are a good many services, which have a different method for its use. So, it is difficult to implement an agent which is able ab initio to use various services. Therefore, we define the *service program* and the *client program* as a pair.

The service provider creates a service program and a client program pair, and discloses a *public policy* in his public zone. A public policy consists of the client program and service attributes. Service attributes consist of a *description* for the explanation of the service, *access\_info* which is information necessary for the service realization, *usage* for showing the purpose of *access\_info* information use

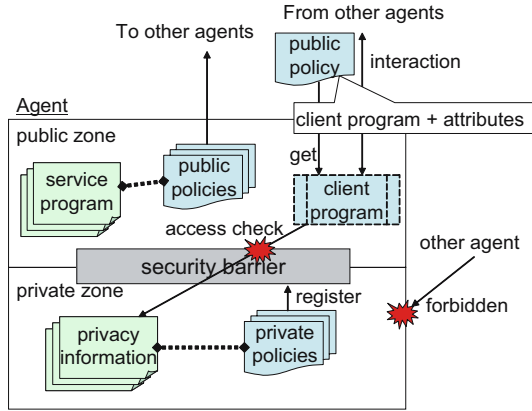


Fig. 2. The Overview of the Public and Private Zone Model

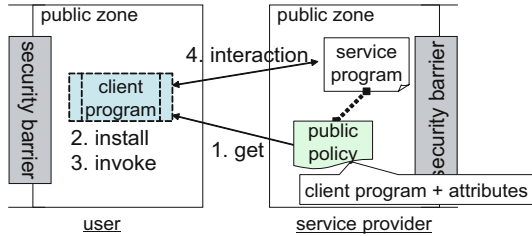


Fig. 3. The Flexible Service Use Mechanism

and *process* which is the utilization process of *access\_info*. A user agent acquires a client program from a service provider agent and invokes it in his public zone. Then the service is actualized through communications, guided by the client program, between the service provider agent and the user agent (Fig. 3). In this way, a user agent can make use of various services without the implementation of explicit methods for the use of the various services.

**The Private Zone:** The private zone manages private information. An agent cannot directly access private information, but must access it through the public zone. Private information has a *privacy policy* which consists of *permission* for specifying access conditions, *allowable\_partner* for specifying communications allowed to a program which accesses private information, *allowable\_usage* for specifying permitted purposes of private information use, and *trusted\_prg* for specifying trusted programs related with *allowable\_usage*. Privacy policies are registered with the security barrier.

When a user wishes to use a service, the user agent acquires the public policy from the service provider agent and invokes its client program. Then, if the client program tries to access private information, the security barrier checks the access by *permission*. If the access is allowed, the security barrier returns

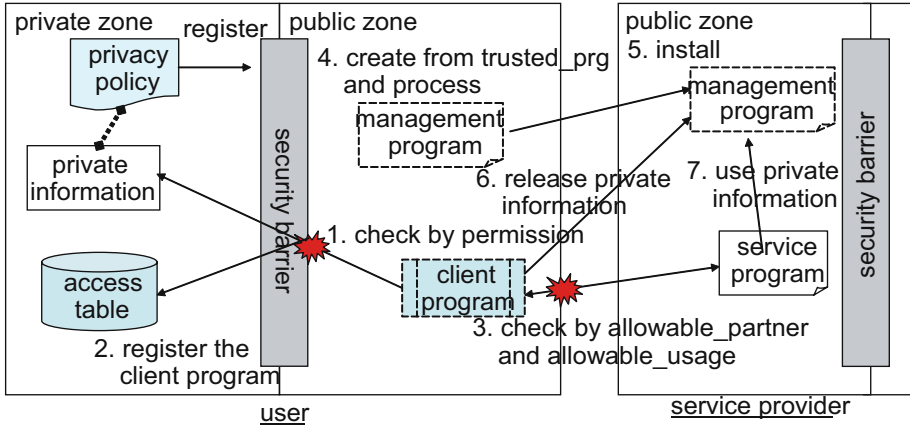


Fig. 4. The Protection of Private Information by the Security Barrier

its value and registers the client program in the access-table; if it is refused, an `IllegalAccessException` happens. After that, the security barrier monitors the communication of the client program registered in the access-table. When the client program communicates with other agents, the security barrier compares *usage* of the client program with *allowable\_partner* and *allowable\_usage* of the privacy policy. If it is refused, an `IllegalCommunicationException` happens; if it is allowed, the security barrier creates a *management program*, as a trusted program, from *process* of the public policy and *trusted\_prg* of the privacy policy. And the user agent sends the management program to the service provider agent. After that, the user agent sends the private information (which is encrypted, not raw data) to the management program (Fig. 4). The service provider agent invokes the management program with this private information. In this manner, the user agent protects private information by restricting the access to private information and by requiring the use of the management program for the control of private information use.

### 3.3 A Simple Application Scenario

We show an example in which a user purchases a product. In the example, the user agent has private credit card information, and the service provider agent provides a product purchase service. The privacy policy of the credit card information is

```
permission=read_only, allowable_partner=publisher_only
allowable_usage={payment:trusted_prg_only}
trusted_prg={payment:trusted_payment_prg}.
```

Service attributes in the public policy are

```
description="Product Sales", access_info=creditcard
usage={creditcard:payment}, process={payment:payment_process}.
```

```

1: SalesProgram extends ClientProrgam{
2:   List product-list; // the product list
3:   Agent owner; // the publisher of this program
4:   public void main(){
5:     display(product-list);
6:     product = a product the user selected
7:     creditcard = accessToPrivateResource("creditcard");
8:     send(owner, {creditcard, price(product)}, process("payment"));
9:   }
10:}

```

**Fig. 5.** The Client Program of the Product Purchase Service

And the client program for the product purchase service is shown in Fig. 5.

Then a user purchases a product as follows.

1. A user agent acquires the public policy of a product purchase service from the service provider agent.
2. The user agent confirms whether the user can use it or not by the *description* and *access\_info* of the public policy. Here, since the agent has credit card information to which access is allowed, the agent shows the product purchase service to the user.
3. When the user wishes to use the product purchase service (e.g. the user clicks "Product Sales" displayed on his terminal), the user agent invokes its client program. Then, the product list is shown to the user at line 5 in Fig. 5.
4. If the user selects a product, the client program tries to access the credit card information at line 7. Here, the access is read\_only, since *permission* is read\_only.
5. The client program tries to send the credit card information to the service provider agent for a payment of the product at line 8. To communicate with the publisher of the client program and to use the credit card information to pay for products are allowed by the *allowable\_partner* and *allowable\_usage*. But the credit card information is only allowed to be used by trusted programs. Therefore, the user agent creates a management program from the payment\_process of *process* and the trusted\_payment\_prg of *trusted\_prg* and sends it to the service provider agent.
6. The user agent sends the credit card information to the management program which was sent at step 5. Then, the service provider agent receives the payment by invoking the management program.

As shown above, the user agent confirms whether the service can be used or not at step 2, and restricts the access to the private information at step 4. Moreover, the user agent monitors the communication of the client program and allows the use of private information only by the trusted program (the management program). In this manner, user agents prevent the service provider agents from using private information for purposes which users do not desire.

## 4 Requirements for the Realization of Our Proposed Model

We have proposed a model for protecting private information, but we must overcome some challenges to realize our model. In this section, we focus on those challenges and discuss their resolvability.

### 4.1 A Method of Creating the Management Program

In our model, a user agent must create a management program from *process* and *trusted\_prg*. Here, an issue is how a management program can be created from *process* and *trusted\_prg*. We suppose this can be realized by combining programs prepared in advance. For example, a user agent analyzes *process* of the public policy and extracts the part which uses private information. After that, the user agent replaces this part with a program specified in *trusted\_prg*.

Consider a product purchase service. The product purchase service requires a credit-card payment and we assume its *process* is shown in Fig. 6. Here, a user agent may not be able to trust *payment-prg* at line 2. Therefore, the user agent replaces line 2 by *trusted-payment-prg* specified in *trusted\_prg*. In this way, a user agent creates a management program from *process* and *trusted\_prg*.

### 4.2 Protection of Management Programs

By compelling a service provider agent to use a management program, user agents protect private information. But if the service provider agent can rewrite the management program, the user agent cannot trust it any longer. Therefore, it is necessary to prevent the rewriting of management programs.

The easiest way to achieve this is to assume the use of anti-tampering devices. We implement public-key cryptography on anti-tampering devices, in which has a public and a secret key pair certified by a certificate authority. A service provider agent opens the public key to the public. A user agent encrypts the management program and private information with the public key and sends the encrypted them to the anti-tampering device of the service provider agent. The anti-tampering device decrypts the encrypted data using the private key and invokes their decrypted versions. In this manner, we can prevent the rewriting of a management program. But this requires that anti-tampering devices be installed in each service provider agent.

Another approach is to make the analysis of a management program difficult for a service provider agent. Mobile cryptography [5] and software obfuscation [1]

```

1 : purchase-program(creditcard, price){
2 :   payment-prg(creditcard, price); -----+
3 :   send a product;                          | replace
4 : }                                           |
                                           trusted-payment-prg(creditcard, price); <--+
    
```

**Fig. 6.** An Example of the Conversion of a Non-trusted Program into a Trusted One



are applicable technologies for this purpose. Mobile cryptography allows direct computations without decryption on encrypted functions. However, it is applicable only to polynomial functions and rational functions. Software obfuscation is a technique which converts a program into another program with a similar behaviour, but which is more difficult to analyze. However, its reliability and evaluation methods are not established.

The purpose of our model is to protect private information, not programs. In other words, a user agent must protect the management program only when private information is exposed to danger. For example, suppose that a management program deletes private information after the process has completed. Then private information can be protected, if it is possible to protect the management program until its process finishes. In this case, we will be able to use software obfuscation to protect the management program, even if its reliability and evaluation methods are not established.

Also, it may be allowed to interact with the user during management program execution and/or to release pieces of a management program little by little as the program progresses. Then it will be easier to protect the management program from service provider agents.

### 4.3 Confirmation of Management Programs

A service provider agent makes use of private information through a management program which is sent from the user agent. If the management program behaves in ways that are undesirable to the service provider agent, the program has no value for the service provider. Accordingly, a service provider agent has to be able to confirm the substance of the management program.

It is difficult to analyze a management program with no information. But a service provider agent knows the workflow of the management program by *process* and asks replaced parts from the user agent. Then, it may be possible to confirm whether the replaced parts work well or not. For example, a service provider agent may be able to confirm that the replaced part in Fig. 6 is warranted by the credit card company. Consequently, a service provider agent can rely on the management program.

Also, as mentioned at Sect. 4.2, a management program is converted to a program (obfuscated program) which is difficult to analyze. So that, it is difficult to analyze the obfuscated program. Here, remember that our purpose is the protection of private information, not programs. Therefore, after finishing the process which requires private information, it is permissible to make the obfuscated program clear. Then the service provider agent will be able to confirm the de-obfuscated program. So, we suppose it is possible to ensure both the protection and confirmation of the management program.

### 4.4 Other Requirements

We must consider the possibility that a management program may be 'malware'. Therefore, we need to protect a service provider agent from malicious manage-

ment programs. Also, to create private policies will be burdensome for users. Therefore, we have to consider who creates privacy policies. If information is supplied from other agents, these agents may be able to offer the privacy policy together with a private information.

## 5 Conclusions

This paper introduced a model for protecting private information and discussed its realizability. In our model, users and service providers are represented as agents who have a public zone and a private zone. The public zone is a freely accessible space for the realization of flexible service use. The private zone is a space for protecting private information by restricting access to private information and by the control of the communications of the client program which accesses private information. When a user needs to send private information to a service provider agent, the user agent compels the service provider agent to make use of private information through user's trusted program. In this manner, the user agent prevents the service provider agent from using private information for purposes which are undesirable to the user.

**Acknowledgements.** This research has been supported by the Telecommunications Advancement Foundation and Strategic Information and Communications R&D Promotion Programme under grant 052310008.

## References

1. D. Aucsmith, and G. Fraunke. Tamper Resistant Software: An Implementation. In *Proc. of International Workshop on Information Hiding*, LNCS 1174, pp. 317–333, 1996.
2. T. Iwao, Y. Wada, M. Okada, and M. Amamiya. A Framework for the Exchange and Installation of Protocols in a Multi-Agent System. In *Proc. of Cooperative Information Agents 2001*, LNCS 2182, pp. 211–222, 2001.
3. C. Pearce, P. Bertok, and R.V. Schyndel. Protecting Consumer Data in Composite Web Services. In *Proc. of 20th IFIP International Information Security Conference*, pp. 19–34, 2005.
4. P3P project. <http://www.w3.org/P3P>.
5. T. Sander, and C. Tschudin. Protecting Mobile Agents Against Malicious Hosts. In *Mobile Agents and Security*, LNCS 1419, pp. 44–60, 1998.
6. K. Takahashi, S. Amamiya, and M. Amamiya. A Model for Flexible Service Use and Secure Resource Management. In *Advances in Grid Computing - EGC 2005*, LNCS 3470, pp. 1143–1153, 2005.
7. The EPAL 1.1. <http://www.zurich.ibm.com/security/enterprise-privacy/epal/>.
8. G. Zhong, S. Amamiya, K. Takahashi, T. Mine, and M. Amamiya. The Design and Implementation of KODAMA System. In *IEICE Transactions INF.& SYST.*, Vol. E85-D, No. 4, pp. 637–646, 2002.

# Performance Analysis of IP Micro-mobility Protocols in Single and Simultaneous Movements Scenario

Giuseppe De Marco<sup>1</sup>, S. Loreto<sup>2</sup>, and Leonard Barolli<sup>1</sup>

<sup>1</sup> Fukuoka Institute of Technology,  
Department of Information and Communication Engineering,  
3-30-1 Wajiro-Higashi, Higashi-ku, Fukuoka, Japan  
{demarco, barolli}@fit.ac.jp

<sup>2</sup> COMICS Lab, Dipartimento di Informatica e Sistemistica,  
University of Naples "Federico II", via Claudio 21, 80125, Napoli, Italy  
salvatore.loreto@ieee.org

**Abstract.** The micro-mobility is an important aspect in ubiquitous communications, where the applications are anywhere and used anytime. In this paper, we analyze two solutions for IP micro-mobility. The first one is based on the Stream Control Transmission Protocol, which allows the dynamic address configuration of an association by means of the AS-CONF messages. The second one is based on Session Initiation Protocol, which is the most popular protocol for multimedia communication over IP networks. Both in the single and simultaneous movements case, we show that for the SCTP solution, there is space for the further optimization of the handoff latency by using slight changes to the protocol. For the SIP solution, we show that for a correct and fast handoff, the SIP Server should be statefull.

## 1 Introduction

Traditionally, the Mobile IP protocol (MIP) has been proposed as network layer (L3) protocol in order to maintain the IP address of mobile node (MN) anywhere it is located [1]. At the application layer, the Session Initiation Protocol (SIP) can also be used to handle MNs [2] [3]. If SIP is used as mobility management, all other applications, like Web and FTP, should be modified in order to use the SIP functionalities. Both in SIP and MIP, the network must employ dedicated servers: MIP Home Agents (HA) and Foreign Agents (FA) and/or SIP Servers<sup>1</sup>. In order to reduce delays and packet losses due to the handoff process, a lot of variants have been proposed for MIP. An example is the hierarchical MIP, H-MIP and its variants. In general, in all the proposals, the network should be augmented with additional components, like Mobility Anchor Point in MIP, or

---

<sup>1</sup> SIP Server can be: Registrar Server, Proxy Server, and so on. We use the general word SIP Server by assuming that a dedicated function is inserted in the SIP Server whenever it is required.

MIP Location Registrar (MIP-LR). These proposals are far from the end-to-end approach of the current Internet design. In this paper, we compare the SIP based solution against the transport layer based solution to the micro-mobility. As transport layer we use Stream Control Transmission Protocol (SCTP) [4]. To the best knowledge of the authors, this is the first attempt to evaluate if SIP and SCTP can be jointly used for single and simultaneous movement of MNs, without introducing neither additional components in the network nor redundant operations, as we discuss in the following. In Sect. 2, we show different types of architecture of micro-mobility by means of a general taxonomy of mobility functions. In Sect. 3, we analyze the latency caused by handoff process for the case of reliable transmission along with possible optimizations.

## 2 Architectures

To better clarify the IP micro-mobility problems, we use a taxonomy based on the following functions:

- The identification function,  $f : x\text{-Id} \rightarrow x\text{-loc}$ , which is used for example when an originating entity (e.g. a correspondent node CN or another MN) wants to communicate with the MN (data are transmitted to or received from it). The originating entity knows the identity of the destination,  $x\text{-Id}$  (e.g.  $x\text{-Id} = \langle \text{giuseppe@sip.fit.ac.jp} \rangle$ ), and needs its locator,  $x\text{-loc}$ .
- The localization and forwarding function,  $g : x\text{-loc} \leftrightarrow \text{IP}$ , which maps the  $x$ 's locator to a visible IP address and viceversa. The  $g$ -function is invoked in order to localize the destination IP address, which can vary over time, e.g. whenever an L3 movement takes place in the case of MN travelling among different subnetworks. In general,  $g(\cdot)$  is not injective, that is  $g$  can have multiple IP addresses associated to its locator.
- The updating function  $u : (x\text{-loc}, \text{IP}_{\text{old}}) \rightarrow (x\text{-loc}, \text{IP}_{\text{new}})$  which is usually executed by MN when its IP address has to be changed. The function will also update the MN's  $\text{IP}_{\text{old}}$  address at the locator agent, e.g. the SIP server.

For example, in MIP,  $f(x\text{-Id}) = \text{IP}_{\text{Home}}$ , that is the *caller* knows the Home IP address of the destination<sup>2</sup>. Then, the HA performs  $g(\text{IP}_{\text{Home}}) = \text{MN}_{\text{CoA}}$ , that is the HA will forward packets towards the current Care of Address (CoA),  $\text{MN}_{\text{CoA}}$ , of the MN. SCTP can be used for the  $u$  function without the support of dedicated location servers as HA and FA in MIP. In fact, SCTP can dynamically add multiple IP addresses during the communication, or association in SCTP terminology, by means of control messages called Association Configuration (AS-CONF) [6] [7].

### 2.1 MIP+SIP

In this case, for VoIP applications, the network is redundant, because both SIP Server and MIP-HA perform the same localization function,  $f$ . This fact has

<sup>2</sup> We use the words *caller* and originating entity as synonymous, although the former word is suitable for multimedia communication. For other details see [5].

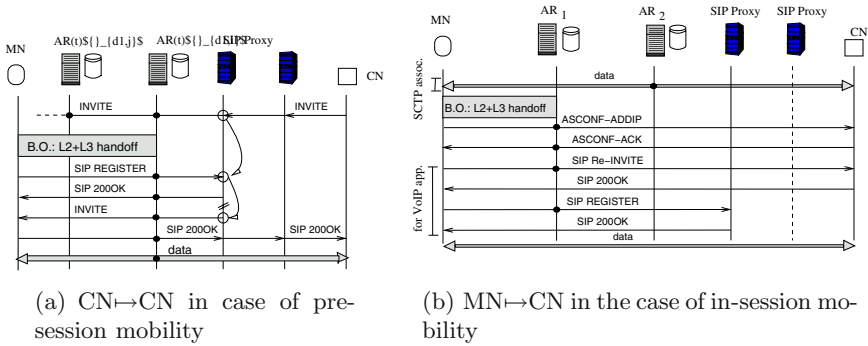


Fig. 1. Mobility in the case of single movement

been observed also in [8], where the authors proposed to use SIP registration whenever it is necessary. For non-VoIP applications, the function  $f$ , if required, is executed by standard mechanisms like DNS. Generally to locate the MN, a SIP Server should be employed. Let us note that if route optimization is not available, MIP does also the function  $g$  (i.e. the packets forwarding), increasing the overhead<sup>3</sup>, the delays and packet losses, both during the handoff and during the communication. In MIPv6, we can avoid the triangulation problem by means of the direct Return Routability Procedure (RRP) to the CN. Decapsulation is not strictly needed because the CoA address is globally routable. We suppose that the MN can acquire a new CoA by means of a DAD CacheServer, which performs in advance the DAD procedure [9].

## 2.2 SCTP+IPv6+SIP

**Single Movement.** In this architecture, the SIP server is aware of the most updated address of the MN. Whenever the MN enters a new subnetwork, an L2/L3 blackout (BO) takes place, as shown in Fig. 1, and the MN acquires a new CoA by means of IPv6 address auto-configuration. In Fig.1-a, the CN initiates the communication (VoIP application) towards the MN by means of the SIP INVITE message. The CN performs the function  $f$  by querying its SIP Server (or the SIP server of the MN if both peers belong to the same domain). Since the location stored in the MN’s SIP server could be not yet updated, the SIP Server should immediately send a pending INVITE whenever it receives a REGISTER message from the interested MN. In fact, usually the REGISTER message is sent to the SIP Server when the SIP application is made aware of a change of the IP address<sup>4</sup>. The alternative is to wait for timers expiration which will delay the call setup. This could be a serious problem if the MN handoff rate

<sup>3</sup> Due to packets decapsulation and triangular forwarding.

<sup>4</sup> This requires dedicated support from the kernel sockets.

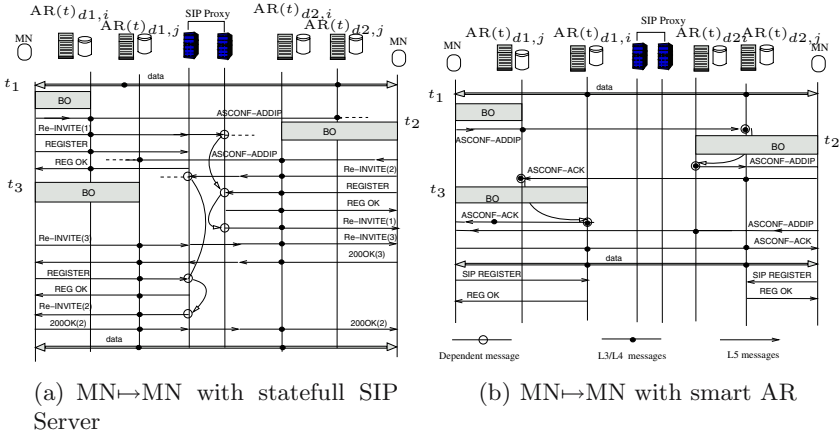


Fig. 2. Mobility in the case of simultaneous movements

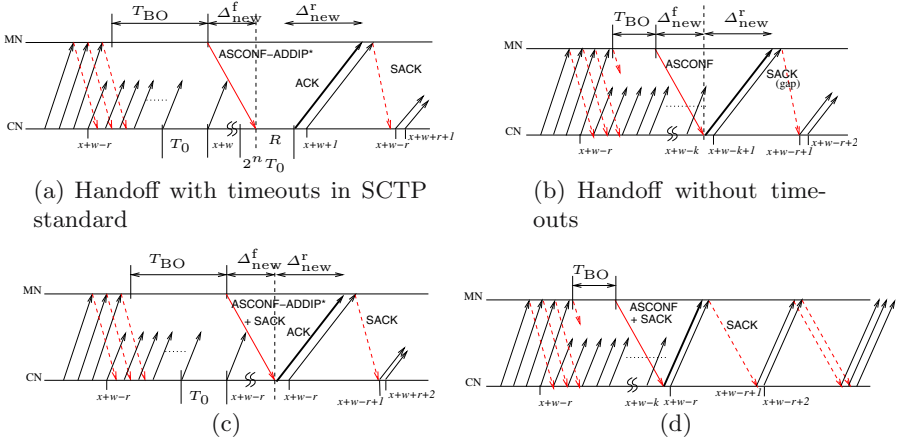
is high. Accordingly, the SIP Server should be statefull. In both cases, for pre-session mobility the SCTP does not play a crucial role. On the other hand, if the MN starts non-VoIP communication towards the CN, as in Fig. 1-b, the SCTP transport will manage the movements of the MN by means of ASCONF-ADDIP and ASCONF-ACK messages. That is, the SCTP performs the  $u$  function inside the MN with  $x-loc = CN_{IP}$ . No other components are needed.

In this architectures (SIP+SCTP+IPv6) redundant registrations are not present. The additional components are: The SIP servers, and the AR supporting IPv6 which are always present in (wireless) subnetwork. One could observe that a full SIP architecture could be enough. But for non-VoIP applications, the function  $f, g$  could not be performed because the CN could not have neither a SIP-based domain nor a SIP-compliant application.

**Simultaneous Movements.** We suppose that the MN’s locations are registered at the SIP Server. For example, suppose that at time  $t_1$  the MN enters a BO phase, and after the L2/L3 handoff, the MN is able to forward and receive packets from  $AR(t_1)_{d1,i}$ , as in Fig. 2-a-b, that is the  $i$ -th AR visited at time  $t_1$  and belonging to domain  $d1$ . At time  $t_2$  the MN in domain 2 moves towards the subnetwork of  $AR(t_2)_{d2,i}$ , and at time  $t_3$  the MN in domain 1 moves towards  $AR(t_3)_{d1,j}$ . In other words, the MNs can travel along different subnetworks,  $i \neq j$ , within the same domain,  $d1, d2$ .

**Full SIP Solution**

The first solution requires the use of a Re-INVITE message, as in Fig. 2-a, because the ASCONF-ADDIP is not able to perform neither the  $f$  function nor  $u$  function because the ASCONF uses an invalid destination address, that is the IP address of the correspondent peer before its movement. In this case, the use of statefull SIP server is mandatory in order to decrease handoff latency and avoid possible signalling incorrectness, as also pointed out in [10].



**Fig. 3.** Loss recovery for different management of ASCONF-ADDIP: case of standard SCTP (a)-(b); in (c), the "Version 2" of Algorithm 2 with timeouts, (d) without timeouts

In Fig. 2-b, we show the other solution which does not require the Re-INVITE message and avoid possible breaking of the communication. We suppose that in every domain the  $AR(\cdot)_{dk,i}$ ,  $k = 1, 2$ , is smart, that is it can store for a certain period of time the new location of the MN just departed. For example, the AR can store an addresses cache for every MN, whose entries being the available addresses for that MN along with their state. Whenever the MN leaves the subnetwork  $AR(\cdot)_{dk,i}$ , the cache for that MN will change the state of the old address, e.g. from a "Preferred" state to a "Deprecated" one<sup>5</sup>, and a new "Preferred" address is being added into the cache, i.e. the MN's address acquired in the subnetwork  $AR(\cdot)_{dk,j}$ . By this way, the  $AR(\cdot)_{dk,i}$  can forward packets whose destination is "Deprecated" to the proper "Preferred" address, e.g. from  $AR(\cdot)_{dk,i}$  to  $AR(\cdot)_{dk,j}$ . We do not enter in the detail of the protocol, but only note that this mechanism could be coupled with the DADCache Server discussed previously.

### 3 Performance Analysis

In this section, we analyze the handoff latency for the mobility schemes discussed above. We define latency as the time interval between the last transmission time on the old path and the transmission time of the first new packet on the new path. Firstly, we note that in wired communication the SCTP protocol receives the ASCONF-ADDIP message from the primary path (i.e. the old path in our scenario) and the transmission can advance without interruption. In wireless transmission, the ASCONF message cannot be sent on the primary path because this path is unavailable unless the wireless NIC allows multiple communication

<sup>5</sup> This mechanism is similar to the addressing scheme of IPv6.

---

1 Algorithm at the sender

---

```

for each received ASCONF-ADDIP
# Pj=new j - th path, ( )i=variables on the previous path
  if (set_primary) & (gap==null) stop all timers;
  # Recovery phase in slow start
  if version1: send(highest(TSNi)) on Pj;
  if version2: send(OutstandingTSNi) on Pj;

```

---

with more than one AP. Accordingly, packet losses and timeouts can occur. Moreover, the transmission over an unknown path should begin with the slow start phase of the congestion control, even if there are enough packets to trigger a Fast Recovery (FR) by sending all the remaining packets over the new path. The solution in [11] respects this constraint, while the authors in [12], without any analysis, suggest to use half of the congestion window on the previous path as the first value of congestion window on the new path. We use slow start phase constraint.

In the case of reliable transmission, the latency is made of two terms: The first one is the delay the sender waits until the arrival of the ASCONF message; the second one is the time of packet loss recovery, as shown in Fig. 3-a for standard SCTP. If no timeout occur, we have re-transmission according with the congestion control as in Fig. 3-b. We now analyze the latency. For the new path, the transmission delays of the MN-CN and CN-MN path are  $\Delta_{new}^f$  and  $\Delta_{new}^r$ , respectively. The BO phase lasts in  $T_{BO}$  seconds. The number of timeouts during the handover is a random variable,  $N$ ; similarly, the number of lost packets is  $L$ . For standard SCTP, we note that if the ASCONF-ADDIP message arrives after a timer is started, the protocol sends the next packet after a time  $R$  until the expiration of the timer, as in Fig. 3-a. In this case, the mean value of latency for  $n$  timeouts is:

$$E\{T_{H_n}\} = E\{T_{BO}\} + E\{R\} + E\{L\} + \Delta_{new}^f + \Delta_{new}^r, \tag{1}$$

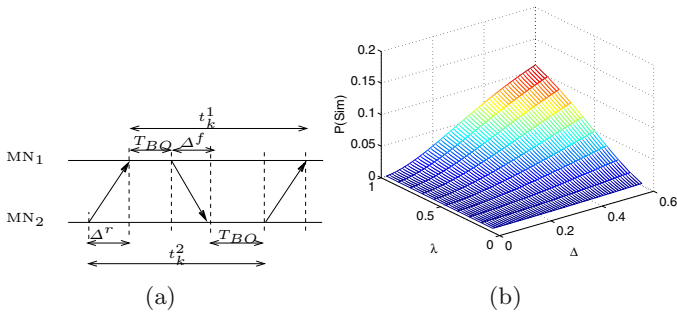
where we suppose  $\Delta^{f,r}$  being deterministic values, and

$$E\{R\} = \sum_{n=1}^{n_{max}} 2^{n-1} T_0 P\{N = n\} \tag{2}$$

$$E\{L\} = \sum_{l=1}^{l_{max}} RTT [ \lceil \log_2 l \rceil + 1 ] P\{L = l\}, \tag{3}$$

where we took into account the slow start phase;  $RTT$  is the round-trip time of the new path. In general, for values of the BO phase and the one-way delays under  $500ms$  and for  $T_0 = 3s$ , we have  $P\{N = 1\} = 1$  and  $P\{N > 1\} = 0$ , that is  $E\{R\} = T_0$ . However, if one accounts also the delays of DAD procedures, the BO phase can be much longer and the probability of more than one timeout increases. The distribution of  $L$  in the sum of (3) depends on the congestion





**Fig. 4.** (a) Critical times in simultaneous movements and (b) probability  $p_s$

window size just before the BO phase and the duration of L3 handoff, i.e. until the arrival of ASCONF message. We suppose that the distribution of  $L$  is known, although it is difficult to derive a closed form formula. If we use the "version1" of Algorithm 1, we obtain:

$$E\{L\} = \sum_{l=1}^{l_{max}} RTT [\lceil \log_2 l \rceil + 1] P\{L = l\},$$

which is the solution of [11]. In that work the packet with  $TSN = x + w + 1$  is called probe packet, where  $w$  is the congestion window before the handoff. However, in that work is not clear if the ASCONF-ACK is sent or not, what is required by the draft document on ASCONF messages [6]. . In our further optimization, we use the "version2" of Algorithm 2, and then we have:

$$E\{L\} = \sum_1^{l_{max}} RTT [\lceil \log_2 l \rceil] P\{L = l\},$$

that is, we can reduce the loss recovery time by one  $RTT$ . This is possible because the "version2" of the algorithm says that if the ASCONF-ADDIP intends to change the primary path immediately and if the SACK, bundled in or following the ASCONF, contains a null gap, then the old path is no longer available and the outstanding packets are surely lost. Then, the go-back- $n$  procedure can be promptly activated. This situation is depicted in Fig. 3-d,e. We note that whatever version of the Algorithm 1 we adopt,  $E\{R\} = 0$ , both for  $T_{H_n}$  and  $T_{H_0}$ . In the case of simultaneous movements, the BO phase perceived by each peer can be longer. In the following, we suppose to use Algorithm 1. For simplicity, we focus on the case of Fig. 2-b and we call MN<sub>1</sub> and MN<sub>2</sub> the MN in domain 1 and 2, respectively; more, we omit the subscript *new* by supposing that the reverse and the forward path delays are equal for old and new path, respectively. We assume the communication delay between the ARs are negligible and for both MNs the delays of BO phase are identically distributed. Let's name with  $Sim$  the event of a deleterious simultaneous movement and with  $P(Sim) = p_s$  its

probability. This event happens when both MNs use an erroneous IP address for the destination of packets. After little algebra, (1) writes as:

$$\begin{aligned} E\{T_{H_n}\} &= p_S E\{T_{H_n}|Sim\} + (1 - p_S) E\{T_{H_n}|\overline{Sim}\} = \\ &= p_S E\{L|Sim\} + (1 - p_S) E\{L|\overline{Sim}\} + \Delta^f + \Delta^r + (1 + p_S) E\{T_{BO}\} \end{aligned} \quad (4)$$

The probability of *Sim* depends on the interval the MNs spend to change the IP address and the rate of the handoff, i.e.  $\lambda_1 = \frac{1}{E\{t^1\}}$  and  $\lambda_2 = \frac{1}{E\{t^2\}}$ . From Fig. 4-a, it is easy to recognize that the critical interval during which a simultaneous movement can happen is  $\Delta^r + \Delta^f$ . In fact, the ADDIP of MN<sub>2</sub> arrives at MN<sub>1</sub> in after a time of  $\Delta^r$  and similarly for the ADDIP of MN<sub>1</sub>. Thus, the probability of *Sim* is the probability of an handoff arrival in these intervals. If we suppose that the handoff interarrival times are exponentially distributed with means  $\lambda_1$  and  $\lambda_2$  for MN<sub>1</sub> and MN<sub>2</sub>, respectively, we have:

$$p_S = e^{-\lambda_1 \Delta^f} e^{-\lambda_2 \Delta^r} \lambda_1 \lambda_2 \Delta^f \Delta^r.$$

In Fig. 4-b, we plot an example of this probability, for  $\Delta^{f,r} = \Delta$  and  $\lambda = \lambda_{1,2}$ . As shown, the probability of simultaneous movements is not negligible as the handoff rate and the path delays increase [10]. In general, the handoff rate depends on the size of the radio cell and the velocity of the MN and can be *high* especially in ubiquitous applications like inter-vehicle communications. We observe that  $p_S$  is also the probability of communication drop if the smart ARs are not employed. In the simultaneous movements case,  $E\{L|Sim\}$  is more complicated, because the number of lost packets depends also on the direction of the communication.

For single movements and SIP scenario, as in Fig. 1-b, the latency is as in (1), because the the SIP Re-INVITE and SIP 200OK messages are the equivalent of ASCONF-ADDIP and its ACK. The slight difference would be that the socket needs not to be re-opened. For simultaneous movements, the latency can be longer. In fact, as shown in Fig. 2-a, the MN<sub>2</sub> must wait the arrival of the SIP Re-INVITE message in order to "re-open" the connection with the new IP of MN<sub>1</sub>. This message is sent by SIP Server of MN<sub>2</sub>, after the reception of the REGISTER message: If the SIP proxy is far, the latency will be higher than that of the SCTP case. In other words, the affected parameters are  $\Delta^r$  and  $\Delta^f$ . Accordingly, the latency would be:

$$E\{T_{H_n}\}_{SIP} \geq E\{T_{H_n}\}_{SCTP},$$

with obvious meaning of the subscript.

## 4 Discussion

In this paper, we recognized that some combinations of mobility protocols, like MIP+SIP, introduce redundancy in the network. With a kind of criticism, we saw that if the standard applications, like Web and FTP, are not supported by SIP based domains, the micro-mobility can not be realized with SIP. On

the other hand, transport layer solutions to the micro-mobility problems are possible whenever the protocol permits the dynamic change of the destination IP address, like SCTP with ASCONF messages. This solution does not require SIP domains. In Sect. 2, we analysed a possible optimization of the handoff latency when ASCONF is used: If the ASCONF contains a "set-primary" parameter, the timers should be stopped, otherwise there is a mean term of  $T_0$  (usually set to 3s) in the latency, at least when the congestion window is less than 2 packets. In Sect. 3, we recognized the following results about simultaneous mobility.

- The optimization of ASCONF is much more important, because  $T_{B0}$  can be much longer and then  $E\{R\}$  could be greater than  $T_0$ .
- The use of SCTP requires changes into the ARs (smart AR).
- The use of SIP requires statefull SIP Server
- The more the SIP Servers are *far* from the subnetwork which the MN is moving towards, the more the use of SIP increases the latency of the handoff. We do not illustrate all possible cases because of page constraints.

As a matter for further investigation, we note that subnetworks equipped with the smart AR could solve both problems of SCTP, for the updating function, and SIP, for the decrease of the handoff latency.

## References

1. D.Johnson, C.Perkins, J.Arkko: Mobility Support in IPv6. IETF RFC 3775. (2004)
2. : (IETF Sip Working Group)
3. Nakajima, N., Dutta, A., S.Das, Schulzrinne, H.: Handoff delay analysis and measurement for sip based mobility in ipv6. In: Proceedings of ICC'2003. Volume 26. (2003) 1085–1089
4. R.Stewart, et al.: Stream Control Transmission Protocol. (IETF RFC 2960)
5. Marco, G.D., Loreto, S., L.Barolli: Performance analysis of ip micro-mobility protocols in single and simultaneous movements scenario. Technical report, FIT (2005)
6. Stewart, R., other: Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration. IETF - draft. (2004)
7. Xing, W., Karl, H., Wolisz, A., Müller, H.: M-sctp: Design and implementation of an end-to-end mobility concept. In: Proceeding of 5th Workshop The Internet Challenge: Technology and Applications. (2002)
8. H.Lee, S.W.Lee, Cho, D.H.: Mobility management based on the integration of mobile ip and session initiation protocol in next generation mobile data networks. In: Proceedings of IEEE Vehicular Technology Conference. Volume 3. (2003)
9. Hwang, S.H., Y.H.Han, Hwang, C.S., Min, S.G.: An address configuration and confirmation scheme for seamless mobility support in ipv6 network. In: Proceedings of Wired/Wireless Internet Communications Conference. Volume 2957 of LCNS., Springer-Verlag (2004) 74–86
10. Wong, K., Dutta, A., K.Young, H.Schulzrinne: Managing simultaneous mobility of ip hosts. In: Proceedings of MILCOM'03. Volume 22. (2003) 785–790
11. Chang, M., M. Lee, H.L.Y.H., Park, J.: An enhancement of transport layer approach to mobility support. In: Proceeding of ICOIN'05. LCNS, Springer-Verlag (2005) 864
12. Aydin, I., Seok, W., Shen, C.C.: Cellular sctp: a transport layer approach to internet mobility. (In: Proceedings of IEEE ICCCN'03) 285–290

# HMRP: Hierarchy-Based Multipath Routing Protocol for Wireless Sensor Networks

Ying-Hong Wang<sup>1</sup>, Hung-Jen Mao<sup>1</sup>, Chih-Hsiao Tsai<sup>1</sup>, and Chih-Chieh Chuang<sup>2</sup>

<sup>1</sup> Department of Computer Science and Information Engineering,  
TamKang University Tamsui, Taipei, Taiwan, R.O.C  
inhon@mail.tku.edu.tw, 693192238@s93.tku.edu.tw,  
890190092@s90.tku.edu.tw

<sup>2</sup> Computer and Communications Research Laboratories,  
Industrial Technology Research Institute, Tainan City, Taiwan, R.O.C  
twjack@itri.org.tw

**Abstract.** Wireless sensor networks consist of small nodes with sensing, computation, and wireless communications capabilities. The energy efficiency is a key design issue that needs to be enhanced to improve the life span of the network. In this paper, we propose a Hierarchy-Based Multipath Routing Protocol (HMRP) for wireless sensor networks. In HMRP, the network will be constructed to layered-network at first. Based on the layered-network, sensor nodes will have multipath route to sink node through some candidate parent nodes. The simulation results show that our HMRP can improve the lifetime of sensor networks.

## 1 Introduction

Wireless sensor networks consist of hundreds to thousands of sensor nodes that have low-power, processor, and limited memory and radio equipments. Those multi-function sensor nodes will be employed in a wide range of applications. Such like military, battlefield, environment monitoring, and civil all can be applied widely with sensor network. Recently, this technology is developed more quickly because of its amount of costs.

The sensor nodes are responsible for collecting information and returning the data it sensed to the Base Station (or Sink node). An important challenge in the design of these networks is battery energy, which limit the lifetime and quality of the networks. Therefore, in order to prolong the lifetime of sensor networks, it is important to design good routing protocols for wireless sensor networks. In [4, 5], the placement of classical sensors will be predetermined and the topology of the network will be in advance. However, in the case of those methods, sensor nodes on the routing path will deplete their energy very fast because of these fixed paths to transfer the sensed data back to the base station (BS). In the past, sensor nodes forward the data packet to the BS directly. But those sensor nodes will consume their battery quickly, so there are many multihop routing protocols proposed in [1, 3, 6, 7, 8, 9] to forward the data packet back to the BS through other nodes. And hierarchical technologies are proposed in [2, 7, 8, 9]. Because sensor nodes will spend a lot of energies when transmit

and receive messages in wireless sensor networks. Hence, the hierarchical routing is an efficient way to decrease energy consumption with data aggregation and fusion.

In this paper, we present an energy-efficient hierarchical mechanism, termed Hierarchy-Based Multipath Routing Protocol (HMRP). We have designed HMRP with the following goals in mind:

**Scalability.** Since unconstrained scale is an inherent feature of sensor network, the solution has to scale to small or large network size.

**Simplicity.** The sensors have limited computing capability and memory resources; we seek to minimize the number of operation performed and the states maintained at each node.

**System Lifetime.** These networks could function for as long as possible. It may be inconvenient or impossible to recharge node batteries. Therefore, all aspects of the node, from the hardware to the protocols, must be designed to be extremely energy efficient.

This paper is organized as follows. Section 2 explains some related technicalities. Section 3 we propose a hierarchy-Based multipath routing protocol for wireless sensor networks. Next, section 4 we present the simulation results. Finally, We give the conclusions in section 5.

## 2 Related Works

In this section we will introduce several hierarchical protocols. A clustering scheme called Low-Energy Adaptive Clustering Hierarchy (LEACH) is proposed in [2] that employs the technique of randomly rotating the role of a cluster head among all the nodes in the network. The operation of LEACH is organized in rounds where each round consists of a setup phase and a transmission phase. During the setup phase, the network will be separated some clusters and will select a cluster head in each cluster randomly. During the transmission phase, the cluster heads collect data from nodes within their respective clusters and apply data fusion before forwarding them directly to the BS. LEACH provides sensor networks with many good features, such as clustering-based, two roles of sensor nodes. However, it expenses much energy in cluster heads when forward data packets to the BS directly.

Another clustering-based routing protocol is Power Efficient Gathering in Sensor Information Systems (PEGASIS) [1], which is constructed from chain-based by using a greedy algorithm. Each node transmits to and receives from only one of its neighbors. In each round, it will be chosen one node on the chain path to transmit the aggregated data to the BS. To locate the closest neighbor node in PEGASIS, each node uses the signal strength to measure the distance of all neighbor nodes. Furthermore, it requires global information of the network known by each sensor node and this does not scale well where such global knowledge is not easy to obtain.

Since data generated in a sensor network is too much for end-user to process, so data aggregated is required. Power Efficient Data Gathering and Aggregation in

Wireless Sensor Networks [9] is proposed. The framework of this proposal is a minimum spanning tree based network. It assumes the locations of all nodes are known by base station and the routing information is computed using Prim's minimum spanning tree algorithm where BS is the root. In this proposal, each round selects the minimum weighted edge from a vertex in the tree to a vertex not in the tree, and adds that edge to the tree. Sensor nodes will transmit the sensed data to BS via the routing path that is constructed before and it achieves a minimum energy consuming system. Nevertheless, the intermediate nodes will consume their energy quickly. And Hierarchy-Based Anycast Routing (HAR) Protocol for Wireless Sensor Networks [6] is proposed. In this scheme, BS constructs hierarchical tree by finding its child nodes by sending packets (such like CREQ, CREP, CACP, PREQ) and discover their own child nodes in turn and so on. The drawback is sending and receiving too many packets will expense much energy of the network.

In this paper, we propose a hierarchy-based multipath routing protocol (HMRP). In the HMRP scheme, the sensor network is constructed as a layered network at first. Based on the layered, sensor nodes will find their candidate parents and transmit the aggregated data to the BS. By way of those candidate parents, the sensor node will has multipath to reply the data. In HMRP, sensor node can switch the routing path to their candidate parents by turns. Our design can distribute the energy cost in the network more efficiently and prolong the network lifetime.

### 3 HMRP: Hierarchy-Based Multipath Routing Protocol

HMRP forms hierarchical relations by using a *Layer Packet*, where nodes can make autonomous relationship without any centralized control. Our goal is to design a hierarchy-based multipath routing algorithm such that when a node may belong to a certain number of parents, it can choose different parent to forward packet every time.

#### 3.1 Layer Packet Format

The format of the layer packet used in HMRP is shown in Fig. 1. It consists of the HopCount, SourceID and SinkID fields. The HopCount field is the number of nodes between sink node and destination node. The nodes that can receive the radio signal of sink are defined as one-hop nodes. SourceID, the ID of the node that layer packet come from, and due to the HMRP support multiple sink, so the SinkID will indicate which one broadcast the layer packet. The number of bytes for each field is best to determine by the application designer. In this paper, we assume that the identifier of sensor nodes is determined a priori.

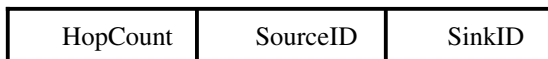


Fig. 1. Layer packet format in HMRP

### 3.2 Network Construction Procedures

The main activities in this phase are hierarchy setup, candidate parents' selection, routing path formation, and information table creation for each node. During each setup phase, we define the hop value of sink is zero, and other sensor nodes are  $\infty$ . The base station first increase the HopCount field by one and broadcasts the layer packet to construct the whole network hop level. During the  $T_{layertime}$ , if sensor node receives many layer packets will compare the HopCount field to choose a less one. And it stores the HopCount, SourceID and SinkID fields of layer packet into HopValue, CandidateParents and SinkID fields of its *Node Information Table (NIT)*.

In Fig. 2, when sensor node 1 receives the layer packets (may come from different sink nodes or other nodes) will choose a node with the less value of HopCount to be its parent. And it will record all this information into NIT. After that, node 1 increase the HopCount field of layer packet by one and broadcast again. At the same time, node 4 will do the similar motion like node 1. Node 6 received two layer packets from node 1 and 4. Hence, its candidate parents are node 1 and 4, the layer packet come from sink A and node 6 is at 2 hop distance from sink A. Every node keeps flooding the layer packet until the network is constructed.

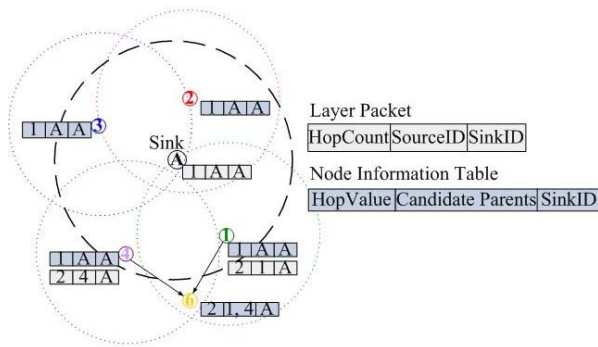
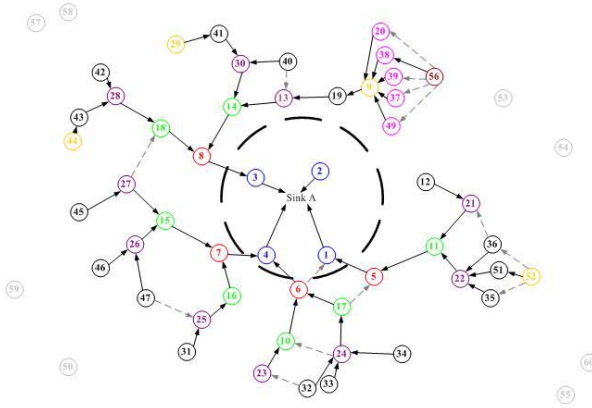


Fig. 2. Network construction flooding

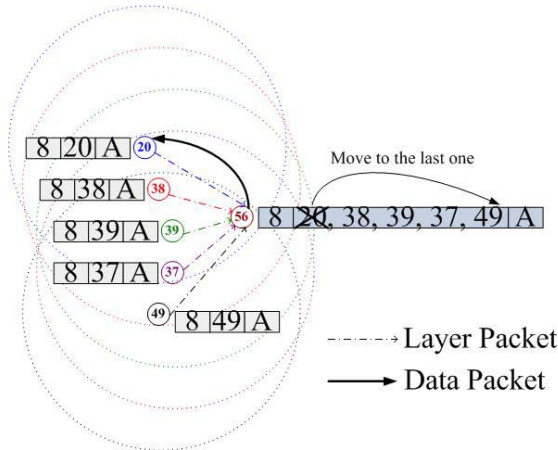
### 3.3 Data Transmission Phase (DTP)

After finish the first phase, the network will be constructed like Fig. 3. Sensor nodes start disseminating the sensed data to the BS via the parent nodes. By using Round Robin method in the CandidateParents field of NIT, when sensor node  $x$  wants to disseminate data packet to sink, it will choose a parent  $p$  from CandidateParents field by turns. And transfer data packet to  $p$ , after  $p$  received this data packet will reply an Acknowledge (ACK) packet to ensure the transmission is successful. If there is no ACK reply from  $p$  after a  $T_{confirm}$  time (this time is very short), node  $x$  will remove  $p$  from its NIT and transfer the data packet to next candidate parent.



**Fig. 3.** Network construction result

For example: in Fig. 4, node 56 has five candidate parents, 20, 38, 39, 37 and 49. It will choose a parent from CandidateParents field sequent. At first time, node 56 disseminates data packet to parent node 20. If node 20 replies an ACK packet, the NIT of node 56 will move the node 20 to last position of CandidateParents field. Otherwise, node 20 will be deleted from the NIT, because its energy may run out or it is broken. Each node does the same motion like node 56 until the data packet reach the sink node.



**Fig. 4.** Data dissemination

Relying on this manner, there are many paths can forward data packet to sink. And sensor node uses different path every time can extend the lifetime of network system.



### 3.4 HMRP Advantages

HMRP employs hierarchical concept to construct whole sensor network. Each sensor node (involve sink node) only broadcast the layer packet once and maintain its own NIT. When sensor node disseminates data packet, it only need to know which parent node to transfer, don't need to maintain the whole path information. This can reduce the overhead of sensor node. Although HMRP has to computing some information to record in the NIT of sensor node, but the energy expense is less than transmit and receive. Furthermore, HMRP support multipath data forwarding, not use the fixed path. So the energy consume will be distributed and the lifetime of network will be prolonged. Finally, HMRP can allow multiple sink nodes.

## 4 Simulation Results

In our analysis, we use the first order radio model discussed in [2, 8]. The transmit and receive energy costs for the transfer of a  $k$ -bit data message between two nodes separated by a distance of  $r$  meters are given by Eqs. 1 and 2, respectively.

$$E_T(k, r) = E_{Tx} k + E_{amp}(r) k \quad (1)$$

$$E_R(k) = E_{Rx} k \quad (2)$$

Where  $E_T(k, r)$  in Eqs. 1 denotes the total energy dissipated in the transmitter of the source node while in Eqs. 2,  $E_R(k)$  represents the energy cost incurred in the receiver of the destination node. The parameters  $E_{Tx}$  and  $E_{Rx}$  are per bit energy dissipation for transmission and reception.  $E_{amp}(r)$  is the energy required by the transmit amplifier to maintain an acceptable radio in order to transfer data message reliably. We use the free-space propagation model and the transmit amplifier  $E_{amp}(r)$  is given by Eqs. 3.

$$E_{amp}(r) = \epsilon_{FS} r^2 \quad (3)$$

Where  $\epsilon_{FS}$  denote transmit amplifier parameter. We assumed the same set of parameters in [2, 8] for all experiments throughout this paper:  $E_{Tx} = E_{Rx} = 50$  nJ/bit,  $\epsilon_{FS} = 10$  pJ/b/m<sup>2</sup> and the energy cost for the data aggregation is set as  $E_{DA} = 5$  nJ/b/message.

To evaluate the performance of HMRP, we use the C++ language to run a number of simulations described in the section. We compare the performance with other cluster-based routing protocols such as LEACH, PEGASIS, HAR and PEDAP. The aim is to measure system lifetime and average energy dissipation. We generate networks of 100 m × 100 m that having 500 nodes and we simulate with different topologies. Furthermore, the initial battery of each node is 2J and the number of data frames transmitted for each round is set at 40. In addition, we fix the message size at 500 bytes in all simulations and assume each sensor node has ability to transmit to BS directly.

Fig. 5 shows the system lifetime of those protocols. HMRP has a good lifetime improvement to others. In addition, we assume the system lifetime is defined as the number of rounds for which 75 percent of the nodes still alive. HMRP has improved the lifetime of other protocols: 200% of LEACH, 8% of PEGASIS, 5% of HAR and 14% of PEDAP. The improvement gained through HMRP is further exemplified by the average energy dissipation graph in Fig 6. HMRP performs energy consumed

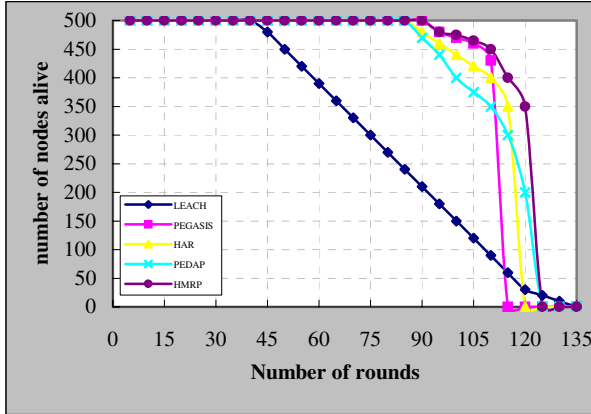


Fig. 5. A comparison of HMRP’s system lifetime with other protocols

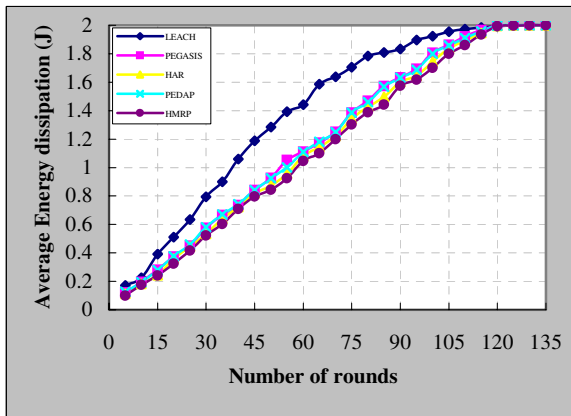


Fig. 6. A comparison of HMRP’s average energy dissipation with other protocols

more efficiently in this plot. On average, HMRP displays a reduction in energy consumption of 35% over LEACH. This is because all cluster heads in LEACH transmit data to the BS directly. However, others alleviate this problem by having only one cluster head node forward the data to the BS. Nevertheless, HMRP still outperform PEGASIS 8% since the distance of neighbors, and HMRP outperforms HAR and PEDAP by 4% and 7%, respectively. Because HMRP spent less cost of energy to construct the hierarchy-structure.

## 5 Conclusions

Energy recourse is limit and is the most important issue to sensor networks. Distributing the load to the nodes has a great impact on system lifetime. In this paper, we pro-

pose a hierarchy-based multipath routing protocol and our main idea is minimizing the path loading of the system by distributing the energy consumption among the nodes. In HMRP, sensor nodes do not to maintain the information of the whole path and they just keep their NITs. We show through the simulation results, HMRP has a better performance than LEACH, PEGASIS, HAR and PEDAP. In addition, it is worth to note that HMRP also supports to multiple base stations (or Sink nodes).

## References

1. Lindsey, S.; Raghavendra, C.S. "PEGASIS: Power Efficient Gathering in Sensor Information System", Aerospace Conference Proceedings, IEEE, pp. 1125-1130, vol 3, 2002.
2. Heinzelman, W.R.; Chandrakasan, .; Balakrishnan, H. "Energy-efficient communication protocol for wireless microsensor networks", System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference, pp. 3005-3014, 2000.
3. Shih-Chang Huang; Rong-Hong Jan. "Energy-Aware, Load Balanced Routing Schemes for Sensor Networks", Parallel and Distributed Systems, 2004. ICPADS 2004. Proceedings. pp. 419-425, 2004.
4. Qiangfeng Jiang; Manivannan, D. "Routing protocols for sensor networks", Consumer Communications and Networking Conference, 2004. CCNC 2004. First IEEE, pp. 93-98, 2004.
5. Al-Karaki, J.N.; Kamal, A.E. "Routing techniques in wireless sensor networks: a survey", Wireless Communications, IEEE, Issue: 6, pp. 6-28. Volume: 11, Dec. 2004.
6. Thepvilojanapong, N.; Tobe, Y.; Sezaki, K. "HAR: Hierarchy-Based Anycast Routing Protocol for Wireless Sensor Networks", Applications and the Internet, 2005. Proceedings. pp. 204-212, 2005.
7. Muruganathan, S.D.; Ma, D.C.F.; Bhasin, R.I.; Fapojuwo, A.O. "A Centralized Energy-Efficient Routing Protocol for Wireless Sensor Networks", communications Magazine, Issue 3, pp. s8-13, IEEE Volume 43, 2005.
8. W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks", IEEE Transactions on Wireless Communications, pp. 660-670, vol 1, no. 4, 2002.
9. Hüseyin Özgür Tan and Ibrahim Körpeoğlu, "Power Efficient Data Gathering and Aggregation in Wireless Sensor Networks", SPECIAL ISSUE: Special section on sensor network technology and sensor data management, pp. 66-71, ACM SIGMOD Record, Volume 32, Issue 4, 2003.

# On Energy-Aware Dynamic Clustering for Hierarchical Sensor Networks\*

Joongheon Kim<sup>1</sup>, Wonjun Lee<sup>1,\*\*</sup>, Eunkyo Kim<sup>2</sup>, Joonmo Kim<sup>3</sup>,  
Choonhwa Lee<sup>4</sup>, Sungjin Kim<sup>1</sup>, and Sooyeon Kim<sup>5</sup>

<sup>1</sup> Department of Computer Science and Engineering, Korea University, Seoul, Korea  
wlee@korea.ac.kr

<sup>2</sup> LG Electronics Institute of Technology, LG Electronics Co., Seoul, Korea

<sup>3</sup> School of Electrical, Electronics, and Computer Engineering,  
Dankook University, Seoul, Korea

<sup>4</sup> College of Information and Communications, Hanyang University, Seoul, Korea

<sup>5</sup> Ubiquitous Computing Lab., IBM-Korea, Seoul, Korea

**Abstract.** This paper proposes an energy-efficient nonlinear programming based dynamic clustering protocol (NLP-DC) unique to sensor networks to reduce the consumption of energy of cluster heads and to prolong the sensor network lifetime. NLP-DC must cover the entire network, which is another basic functionality of topology control. To achieve these goals, NLP-DC dynamically regulates the radius of each cluster for the purpose of minimizing energy consumption of cluster heads while the entire sensor network field is still being covered by each cluster. We verify both *energy-efficiency* and *guarantee of perfect coverage*. Through simulation results, we show that NLP-DC achieves the desired properties.

## 1 Introduction

Recently many research efforts on wireless sensor networks have become one of the most active research activities in wireless communications and networks technologies [1] [2]. Sensors, the main components of wireless sensor network, are deployed over the sensing network fields, and perform the specific tasks with the processing, sensing, and communicating capacities [2]. Due to their limited power source, energy consumption has been concerned as the most critical factor when designing sensor network protocols. Facing this challenge and research issues, several approaches to prolong lifetime of the sensor networks, including clustering schemes and structured schemes with a two-tiered hierarchy, have been investigated. The clustering technology facilitates the distribution of control over the network and enables locality of communications [3]. The two-tiered hierarchical structuring method is an energy-efficient scheme for wireless

---

\* This work was sponsored by SK Telecom, Korea under Contact Number KU-R0405721 to Korea University and by Grant No. R01-2005-000-10267-0 from Korea Science and Engineering Foundation in Ministry of Science and Technology.

\*\* Corresponding author.

sensor networks [4]. It consists of the upper tier for communicating among cluster heads (CHs) and the lower tier for sensing events and transmitting them to CHs. However, in traditional clustering scheme and two-tiered hierarchical structuring scheme, CHs cannot adjust their radius, which leads to inefficiency in terms of energy conservation and network management. If the cluster range is larger than optimal one, a CH consumes more energy than required. On the other hand, a smaller range than necessary results in the entire sensing field not being covered. To fulfill these requirements, we propose a novel clustering algorithm which aims to minimize energy consumption of CHs under the hierarchical structure presented in [4]. Our proposed clustering scheme, NLP-Based Dynamic Clustering (NLP-DC), is able to regulate the cluster radius for energy savings while the entire sensor network field is still being covered by each cluster. Considering the inherent features of NLP-DC, more powerful capability can be given to CHs than other sensor nodes. The NLP-DC could be applied to IEEE 802.15.4/ZigBee-based protocol in the aspect of energy conservation because the features of IEEE 802.15.4/ZigBee-based protocol have the different type of sensor nodes and the various type of topology (i.e., star/mesh/cluster-tree topology) based on clustering [5]. The remainder of this paper is organized as follows. In Section 2, we investigate previous work with focuses on clustering scheme and hierarchical structure scheme in wireless sensor networks. Section 3 describes NLP-Based Dynamic Clustering, proposed scheme in this paper. We prove the guarantee of perfect coverage of NLP-DC via theoretical analysis in Section 4. We evaluate the performance of NLP-DC via simulation in Section 5. Section 6 concludes the paper and presents the direction of our future work.

## 2 Related Work

Several clustering and hierarchical structure schemes aiming to improve energy-efficiency in wireless sensor networks have been proposed. LEACH [6], a protocol architecture for sensor networks that combines the ideas of energy-efficient cluster-based routing with application-specific data aggregation to achieve good performance in terms of system lifetime, latency, and application-perceived quality. As one of the most notable clustering schemes in wireless sensor networks, it preserves limited amount of energy by selecting a CH at random among sensor nodes. In [7], Gupta et al. proposed a two-tiered hierarchical clustering scheme in which a CH with less energy constraint is selected among sensor nodes. Based on this method, it pursues an energy-efficient network management. However, a CH is selected among the sensor nodes as assumed in [6]. FLOC (Fast Local Clustering) [8], provides an adjustable method of cluster radius, but it is limited to one-hop or two-hop distances. The clustering-based topology control scheme [4] consists of two tiers; (1) upper tier for communicating between CHs and (2) lower tier for sensing, processing, and transmitting by sensors. It has similarity on load-balancing concept with NLP-DC. However, its performance depends on each radius of the cluster; as a cluster is increased covering the whole sensor network area, the energy consumption can be increased. In the previous schemes

suggested so far, they have common features that the radius of each cluster head is fixed and mainly concentrated on sensor nodes. On the other hand, our proposed NLP-DC considers a CH to be a device which has more computing power and less energy-constraint. Also, it doesn't need any specific routing protocols for sensor networks since the radius of cluster radius does not concern with the distance estimated by hop count. From the next sections, we give detailed explanations on our proposed NLP-DC, which improves performance in terms of load-balanced efficient energy consumption on CHs.

### 3 NLP-Based Dynamic Clustering

The NLP-DC aims at covering the entire network with the minimum power consumptions of cluster heads by regulating the cluster range. The basic concept of NLP-DC was proposed in [9]. If the cluster range is larger than optimal one, a CH consumes more energy than required. On the other hand, a smaller cluster range than optimal one results in the entire sensing field not being covered.

In NLP-DC, the whole sensor network field is being covered by CHs and the CHs consider their weights assigned by *penalty functions* (e.g., residual energy) and *reward functions* (e.g., priority). For achieving energy efficiency, a sink computes objective functions. Energy-efficient radii for each cluster are calculated based on the objective functions. Followings are what we assume in NLP-DC.

- The architecture of wireless sensor network has two-tiered hierarchical structure. The upper tier consists of CHs and the lower tier consists of sensors.
- A sink knows the position of each CH.

#### 3.1 Initial Phase

In initial phase, the CHs deployed at random construct a triangle to determine a *Dynamic Clustering Controller (DCC)* that is able to minimize energy consumptions of CHs while covering the entire network field as shown in Fig. 1.

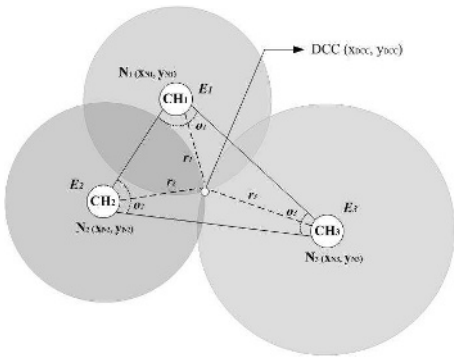


Fig. 1. System model for NLP-based dynamic clustering (NLP-DC)

The distance between DCC and each CH can be estimated as the radius of each cluster. *Delaunay* triangulation [10] [11], which guarantees the construction of an approximate equilateral triangle, is used for constructing a triangle. The construction of equilateral triangles leads to load-balanced energy consumption of each CH. The load-balanced energy consumption can prolong network lifetime.

### 3.2 NLP-Based Iterative Phase

The cluster radii of three CHs can be dynamically controlled by using a DCC as a pivot. The goal is to determine the positions of DCCs which minimize the energy consumption of each CH by finding energy-efficient cluster radii. As shown in Fig. 2., the triangle which is composed by three CHs. If a cluster becomes large, the CH of the cluster will consume more energy because the number of sensor nodes to be controlled will be increased. Therefore, if the overlapping areas of each sector are larger than optimal one, CHs consume more energy than required. Therefore it is important to find the minimized overlapping areas while covering the entire network. The size of overlapping area can be obtained by extracting the size of triangle from the summation of three sectors. By this concept, we can derive the objective function to find the DCC for minimizing the overlapping areas. The areas of sectors can be obtained by

$$S = \frac{1}{2}\theta \cdot r^2. \tag{1}$$

Therefore a DCC is determined by an objective function as the Eq. (2).

$$\begin{aligned} \text{minimize: } f(r_1, r_2, r_3, \theta_1, \theta_2, \theta_3) &= \frac{1}{2} \sum_{k=1}^3 \theta_k \cdot r_k^2 - S_{triangle} \\ \text{s.t. } r_i^2 &= (x_{DCC} - x_i)^2 + (y_{DCC} - y_i)^2 \end{aligned} \tag{2}$$

In Eq. (1) and Eq. (2),  $\theta_k$  denotes the angle value of  $CH_k$ ,  $r_k$  means the distance between  $DCC$  and  $CH_k$ . Also  $S_{triangle}$  is the area of triangle which is comprised by *Delaunay* triangulation. As for an nonlinear programming method to solve the objective functions presented in this paper, we use an *L-BFGS method* [12], one of the most efficient nonlinear programming methods to solve unconstraint

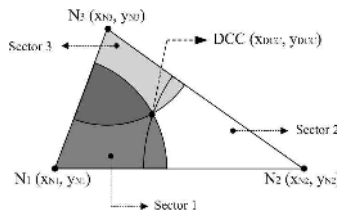


Fig. 2. Conceptual model to minimize overlapping areas

optimization problem. As shown in the Eq. (2), the priorities in each CH do not considered in the objective function. However in certain cases, we need to assign a different weight to each CH. If the CH in certain area is more important than the other CHs, we need to assign higher priority to the CH. Also if events occur in some endemic specific area, the CH in that area must be assigned with a higher priority. By these necessities, the consideration of weight functions is necessary. We can consider *penalty functions* and *reward functions* for the weight functions. If the penalty function of a CH has a large value, the CH must reduce its cluster radius. If the reward function of a CH has a large value, the CH must enlarge its cluster radius. By applying this concept, we can derive the Eq. (3).

**minimize:**  $f(r_1, r_2, r_3, \theta_1, \theta_2, \theta_3, \phi_{1,1}(\vec{x}), \dots, \phi_{m,3}(\vec{x}), \psi_{1,1}(\vec{x}), \dots, \psi_{n,3}(\vec{x}))$

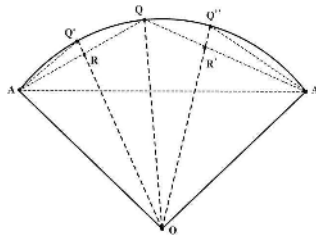
$$= \frac{1}{2} \sum_{k=1}^3 \theta_k \cdot r_k^2 \cdot \prod_{l=1}^m \frac{\phi_{l,k}(\vec{x})}{\frac{1}{3} \sum_{l=1}^3 \phi_{l,i}(\vec{x})} \cdot \prod_{g=1}^n \frac{\frac{1}{\psi_{g,k}(\vec{x})}}{\frac{1}{3} \sum_{i=1}^3 \frac{1}{\psi_{g,i}(\vec{x})}} - S_{triangle} \quad (3)$$

s.t.  $r_i^2 = (x_{DCC} - x_i)^2 + (y_{DCC} - y_i)^2$

The notations of Eq. (3) are the same as the notations of Fig. 1, Fig. 2, Eq. (1), and Eq. (2). In Eq. (3), NLP-DC assigns a weight function to each CH where  $\phi_{i,j}(\vec{x})$  and  $\psi_{i,j}(\vec{x})$  represents a *penalty function* and a *reward function*, respectively. This objective function, Eq. (3), has  $m$  penalty functions and  $n$  reward functions.

### 4 Guarantee of Perfect Coverage

The guarantee of perfect coverage, the second main contribution, will be shown in this section in the aspect of theorem-based theoretical analysis.



**Fig. 3.** Notations for corollary 1

**Theorem 1:**

NLP-DC can guarantee the perfect coverage. That is, the entire sensor network considered lower layer is covered by clusters in the upper layer.





**Corollary 2:**

Sector  $OAA'$  in Fig. 3 covers rectangle  $OAQA'$ , perfectly.

**Proof:**

If we choose any  $Q$  ( $DCC$  in NLP-DC) in the arc  $AA'$ , (1) an isosceles triangle  $AOQ$  is covered by a sector  $AOQ$  (by Corollary 1), (2) an isosceles triangle  $A'OQ$  is covered by a sector  $A'OQ$  (by Corollary 1), (3) a sector  $OAA' =$  a sector  $AOQ +$  a sector  $A'OQ$ , and (4) a rectangle  $OAQA' =$  an isosceles triangle  $AOQ +$  an isosceles triangle  $A'OQ$ . By (1), (2), (3), and (4), a sector  $OAA'$  covers a rectangle  $OAQA'$ , perfectly.

**End of Proof: Corollary 2**

**Corollary 3:**

The triangle constructed by *Delaunay* triangulation in an initial phase, is covered by three rectangles, perfectly.

**Proof:**

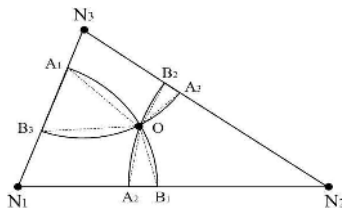
As shown in Fig. 5, when the three rectangles are united, the overlapped areas are generated while the triangle is covered, perfectly. Overlapped areas are triangle  $OA_1B_3$ , triangle  $OA_2B_1$ , and triangle  $OA_3B_2$ . Therefore we prove that the triangle generated in an initial phase is covered by three rectangles.

**End of Proof: Corollary 3**

As proved in Corollary 1, Corollary 2, and Corollary 3, rectangle  $OA_1N_1B_1$  is covered by sector  $A_1N_1B_1$ . And rectangle  $OA_2N_2B_2$  is covered by sector  $A_2N_2B_2$ . And rectangle  $OA_3N_3B_3$  is covered by sector  $A_3N_3B_3$ . Also, triangle  $N_1N_2N_3 =$  rectangle  $OA_1N_1B_1 +$  rectangle  $OA_2N_2B_2 +$  rectangle  $OA_3N_3B_3$ . Then, by the triangle  $N_1N_2N_3$ , *Delaunay* triangle which is constructed in an initial phase, is covered by three rectangles, the triangle  $N_1N_2N_3$  is covered by the three sectors, perfectly. That is,

$$\text{triangle } N_1N_2N_3 \subset \text{three rectangles} \subset \text{three sectors}$$

Therefore, based on these three corollaries, theorem 1 is proved.



**Fig. 5.** Notations for corollary 3

## 5 Simulation Results

A simulation study is conducted to evaluate the performance of NLP-DC. CHs and sensor nodes were randomly deployed. As a simulation setup, we place five CHs randomly in the upper layer. Also, performance evaluation is executed ten times. Our simulations were designed to evaluate the effect of energy-efficiency, one of the main contribution of NLP-DC. We compare NLP-DC against the method that has a fixed cluster radius, named FCR in this paper. In FCR, the CHs in a *Delaunay* triangle has cluster radii which can cover each other and be fixed. In other words, the cluster radius is fixed but it has to be extended to the distance to allow communication among neighbor CHs in FCR. To show the energy-efficiency, we show (1) overlapping areas and (2) residual energy.

### 5.1 Overlapping Areas

The performance of NLP-DC and FCR are compared by measuring the percentage of overlapping areas. We compared FCR and NLP-DC in hierarchical sensor network architecture. As mentioned in the head of this section, we deployed five CHs in upper layer and measured the percentage of overlapping areas.

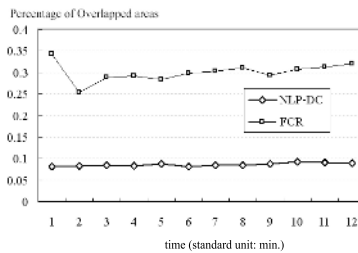


Fig. 6. The percentage of overlapping areas

When  $S$  and  $OA$  represents the size of network field and overlapping areas, respectively, the the percentage of overlapping areas is measured by  $\frac{OA}{S}$ . Fig. 6 plots the percentage of overlapping areas. As shown in Fig. 6, the percentage of overlapping areas of FCR is between 0.25 and 0.35. On the other hands, the percentage of overlapping areas of NLP-DC is 0.08, approximately. Therefore NLP-DC is more efficient than FCR almost from 3.125 times to 4.375 times in the aspect of the percentage of overlapping areas.

### 5.2 Consumed Energy of All Cluster Heads

The performance of NLP-DC and FCR are compared by measuring the percentage of consumed energy of CHs in this subsection.

We also compared FCR and NLP-DC in clustering-based hierarchical sensor network architecture. Fig. 7 presents the percentage of consumed energy of all

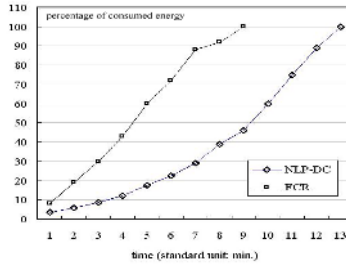


Fig. 7. The percentage of consumed energy of all cluster heads

CHs in upper layer. As shown in Fig. 7, the energy of all CHs of FCR is totally vanished between eight and nine minutes. On the other hands, the energy of all CHs of NLP-DC is vanished completely between twelve and thirteen minutes. Therefore NLP-DC is more energy-efficient than FCR almost from 1.333 times to 1.625 times in the aspect of the percentage of consumed energy of all CHs.

## 6 Conclusions and Future Work

Our proposed scheme, NLP-DC, is able to regulate the cluster radius for energy savings while the entire sensor network field is still being covered totally by each cluster in hierarchical sensor network architecture. To accomplish these objectives, NLP-DC dynamically regulates the radius of each cluster for the purpose of minimizing energy consumption of cluster heads. Therefore there exists two kinds of main contributions in this paper. We show both the 'energy-efficiency' through the design rationale of NLP-DC shown in section 3 and the 'guarantee of perfect coverage' through theorem-based theoretical analysis shown in section 4. Through simulation-based performance evaluation, the novelty on NLP-DC is shown in the aspect of energy-efficiency. More complicated situations in which sensor nodes have mobility must be considered by the emergence of wireless embedded sensors as one of dominant networking technology trends. Our future work includes the development of sensor network protocols suitable for such environments. Based on the proposed algorithm in this paper, we are developing a novel RFID reader anti-collision protocol that minimizes the overlapping areas among clusters by dynamically regulating the cluster radius among RFID readers to optimize the reader collisions. The preliminary results of the research on RFID reader anti-collision using NLP-DC is presented in [13].

## References

1. D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next Century Challenges: Scalable Coordination in Sensor Networks," in Proc. of ACM MobiCom, Seattle, WA, USA, Aug. 1999.
2. I. F. Akyildiz, W. L. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: A Survey", Computer Networks, Elsevier Science, 38(4), pp. 393-422, Mar. 2002.

3. V. Mhatre and C. Rosenberg, "Design Guidelines for Wireless Sensor Networks: Communication, Clustering and Aggregation," *Ad Hoc Networks Journal*, Elsevier Science, 2(1), pp. 45 - 63, 2004.
4. J. Pan, Y. T. Hou, L. Cai, Y. Shi, and S. X. Shen, "Topology Control for Wireless Sensor Networks," in *Proc. of ACM MobiCom*, San Diego, CA, USA, Sep. 2003.
5. J. Zheng and M. J. Lee, "Will IEEE 802.15.4 Make Ubiquitous Networking a Reality?: A Discussion on a Potential Low Power, Low Bit Rate Standard," *IEEE Communications Magazine*, 42(6), pp. 140 - 146, Jun. 2004.
6. W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," *IEEE Transactions on Wireless Communications*, 1(4), pp. 660 - 670, 2002.
7. G. Gupta and M. Younis, "Load-Balanced Clustering of Wireless Sensor Networks," in *Proc. of IEEE ICC*, Achnorage, AK, USA, May 2003.
8. M. Demirbas, A. Arora, V. Mittal, and V. Kulathumani, "Design and Analysis of a Fast Local Clustering Service for Wireless Sensor Networks," in *Proc. of IEEE BROADNETS*, San Jose, CA, USA, Oct. 2004.
9. J. Kim, E. Kim, S. Kim, D. Kim, and W. Lee, "Low-Energy Localized Clustering: An Adaptive Cluster Radius Configuration Scheme for Topology Control in Wireless Sensor Networks," in *Proc. of IEEE VTC*, Stockholm, Sweden, May 2005.
10. M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry: Algorithms and Applications*, 2nd Ed., Springer-Verlag, 2000.
11. F. Aurenhammer, "Voronoi Diagrams - A Survey of a Fundamental Geometric Data Structure," *ACM Computing Surveys*, 23(3), pp. 345 - 405, Sep. 1991.
12. D. C. Liu and J. Nocedal, "On the Limited Memory BFGS Method for Large Scale Optimization," *ACM Mathematical Programming*, 45, pp. 503 - 528, Dec. 1989.
13. J. Kim, W. Lee, J. Yu, J. Myung, E. Kim, and C. Lee, "Effect of Localized Optimal Clustering for Reader Anti-Collision in RFID Networks: Fairness Aspect to the Readers," in *Proc. of IEEE ICCCN*, San Diego, CA, USA, Oct. 2005.

# Neighbor Node Discovery Algorithm for Energy-Efficient Clustering in Ubiquitous Sensor Networks

Ji Young Choi<sup>1</sup>, Chung Gu Kang<sup>1,\*</sup>, Yong Suk Kim<sup>2</sup>, and Kyeong Hur<sup>3</sup>

<sup>1</sup> Department of Radio Communications Engineering, Korea University  
{jychoi2004, ccgkang}@korea.ac.kr

<sup>2</sup> SAMSUNG Advanced Institute of Technology (SAIT), Korea  
yongsuk@samsung.com

<sup>3</sup> Department of Computer Education, Gyeongin National University of Education  
khur@ginue.ac.kr

**Abstract.** Clustering algorithm is an essential element to implement a hierarchical routing protocol, especially for a large-scale wireless sensor network. In this paper, we propose a new type of energy-efficient clustering algorithm, which maximizes the physical distance between cluster head and gateway by a neighbor node discovery mechanism. Furthermore, a slave/master patching scheme is introduced as a useful means of further improving the energy-efficiency. It has been shown that the number of cluster heads can be reduced by as many as 21% as compared with the existing clustering algorithms.

## 1 Introduction

Numerous routing protocols have been developed for discovering and maintaining routes in the large-scale wireless sensor networks (WSNs). A cluster-based hierarchical routing protocol is one of those, especially useful for prolonging the lifetime of WSNs. In a flat routing protocol, multi-hop connectivity is provided to communicate with a base station (BS). In a cluster-based hierarchical routing protocol, however, a sensor field region is divided into several clusters. Each cluster has its own cluster head, while all nodes in a cluster are within a direct radio range of the cluster head. Nodes located within the radio range of more than one cluster heads are designated as gateways, which is used for inter-cluster communication. The data gathered by a sensor node is transmitted to a cluster head, which subsequently relays it to a BS through a hierarchy of cluster heads and gateways. As only cluster heads and gateways participate in the routing process, hierarchical routing tends to be more energy-efficient than a flat routing structure.

Data aggregation is another energy-aware element in WSNs. Once a particular event is detected in a region, it is transmitted to the corresponding cluster head, which usually aggregates the received data before transmitted to the BS. Data aggregation is to combine the data coming from nearby region so as to eliminate the data redundancy for reducing the overall communication load [4]. Several clustering algorithms [1-3, 6-8] are essential elements to realize hierarchical routing and data aggregation.

---

\* Member IEEE.

If more than two sensor nodes in the same cluster transmit at the same time, intra-cluster collision is incurred. Two broad categories of wireless channel access scheme are used for multiple access communication: contention-based and reservation-based. A contention-based scheme such as CSMA-CA protocol with RTS/CTS handshaking signals can be used for intra-cluster communication, i.e., communication between a cluster head and sensor nodes. However, the number of retransmissions is increased with a large number of sensor nodes, incurring unnecessary power consumption. Meanwhile, a reservation-based scheme such as TDMA protocol, allocates unique time slots to each sensor node, can significantly reduce the power consumption in multiple access communication. Furthermore, it allows sensor nodes to turn off their receiver and/or transmitter when they are not scheduled to communicate.

Although TDMA protocol can be more energy-efficient than CSMA-CA protocol, it is hard to coordinate the time slots for inter-cluster communication, especially when a single wireless channel (code) is employed. In other words, intra-cluster communication may be interfered with another inter-cluster communication unless there is some centralized means of coordinating two different levels of communication. Due to the distributed nature of WSN, the centralized TDMA protocol may be too expensive for handling multi-cluster-based communication in the course of hierarchical routing. The different approach is to combine TDMA and CSMA-CA protocols as in HiPERMAC [Grant No. KOR-2005-0046460(2005)], in which TDMA is used for intra-cluster communication while CSMA-CA is used for inter-cluster communication.

As mentioned earlier, cluster heads and gateways must be always turned on since they involve with data aggregation and/or hierarchical routing process while a gateway cannot use the energy-efficient TDMA protocol due to inter-cluster collision. Therefore, one of the most important design objectives is to reduce the number of cluster heads and gateways in the course of clustering process, which can increase the overall lifetime of WSNs. In this paper, we propose a new type of energy-efficient clustering algorithm, in which a simple protocol is designed for counting the number of neighbor nodes.

The remainder of this paper is organized as follows. In Section 2, we summarize operational characteristics of the previous related works. In Section 3, we present details of the proposed clustering algorithm. Simulation results are given to evaluate its performance in Section 4. Finally, Section 5 concludes the paper.

## 2 Related Works

Many clustering algorithms over a single wireless channel (code) have been proposed to choose cluster heads for operation of energy-efficient wireless media access control and hierarchical routing in WSNs. Among those, Lowest ID clustering algorithm [1], also known as identifier-based clustering, proceeds as follows. Each node is assigned a unique ID and it periodically broadcasts a list of its neighbors (including itself). A node which only hears nodes with ID higher than itself becomes a “cluster head.” A node which can hear only one cluster head becomes an “ordinary node.” A node which can hear two or more cluster heads becomes a “gateway” and it lies within the radio range of two or more cluster heads. The gateway nodes are generally used for routing between clusters. Although Lowest ID is one of the most simple and popular

clustering schemes, its drawback is that too many cluster heads and gateways may be generated. Topology Discovery algorithm [3] attempts to reduce the number of cluster heads and gateways by maximizing a physical distance between a cluster head and a gateway. It assumes that propagation delay is proportional to their physical distance between sending and receiving nodes. A cluster head broadcasts a topology discovery request packet and the node which has received it sends back a response to the cluster head. The cluster head measures this propagation delay from each node and elects one which has a longest propagation delay as a gateway. Meanwhile, the gateway elects a cluster head of neighbor cluster in the same way. This same election procedure is repeated until each node is assigned to at least one cluster. The assumption that “propagation delay is proportional to their physical distance between sending and receiving nodes” cannot be guaranteed when multi-path fading is considered in the general wireless channel environment. Furthermore, it may be difficult to distinguish a slight difference of each propagation delay since radio range of WSNs is generally within ten meters.

### 3 Neighbor Node Discovery (NND) Algorithm

We first present the underlying concept of our clustering algorithm and then, illustrate its detailed operation.

#### 3.1 Design Objective and Constraint

We aim at forming the multiple clusters to cover all communicating nodes, while facilitating an energy-efficient operation for the large-scale WSNs. As opposed to the existing protocols with multiple frequency channels, e.g., in terms of orthogonal codes or frequency bands, we consider a system with a single frequency band, which allows for the cost-effective transceiver implementation and flexible network deployment. In general, it is a challenging objective to design the access schemes for both intra-cluster and inter-cluster communication only by using a single frequency band. In our design, a dynamic reservation TDMA scheme is considered for intra-cluster communication as a means of sharing the wireless link in an energy-efficient manner. Meanwhile, a contention-based CSMA-CA scheme is considered for inter-cluster communications as a means of avoiding inter-cluster collision. A superframe is divided into two different periods, one for TDMA and the other for CSMA-CA protocol. Fig. 1 shows the organization of sensor nodes in a hierarchical routing structure, illustrating four different types of nodes found in the course of clustering: cluster head (CH), initial node (IN), ordinary node (ON), and gateway (GW). A cluster is a set of nodes within one-hop range from a CH, which coordinates the TDMA time slots sharing among those nodes within the cluster while providing a link with the gateway nodes for inter-cluster communication. The INs are defined as a set of nodes that have not been covered by any cluster. INs are turned into either CHs, ONs, or GWs in the course of performing the clustering algorithm. CH and GW should be always turned on since they participate in data aggregation and/or routing process. Therefore, they consume more energy than an ON, depleting their battery faster. The overall network lifetime can be prolonged by reducing the number of CHs and GWs. Meanwhile, note that ON



is more energy-efficient than GW since it can turn off its receiver and/or transmitter when it is not scheduled for TDMA time slots. In this case, the network lifetime can be further prolonged by increasing the number of ONs. Our design objective for the multi-cluster routing architecture operating with a single frequency band is to control the numbers of the different types of nodes for increasing the overall energy efficiency, i.e., minimizing the number of CHs and GWs.

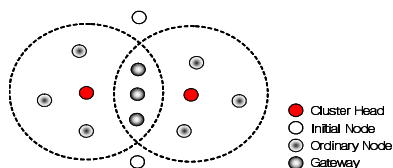


Fig. 1. Types of sensor nodes in a hierarchical routing architecture

### 3.2 Basic Operation

The essence of our approach is to discover neighbor nodes. One of the nodes at the cluster edge is designated as a GW, while a node which is farthest from GW is designated as a CH of neighbor cluster. Basic operation of the NND algorithm consists of 4 different procedures: initial cluster set-up, gateway election, cluster head election, and gateway re-election procedures. All these procedures are combined and repeated until all INs are turned into either CH, GW, or ON. A sequence of these procedures is shown with a flow chart in Fig. 2. The detailed operation is described in the following subsections.

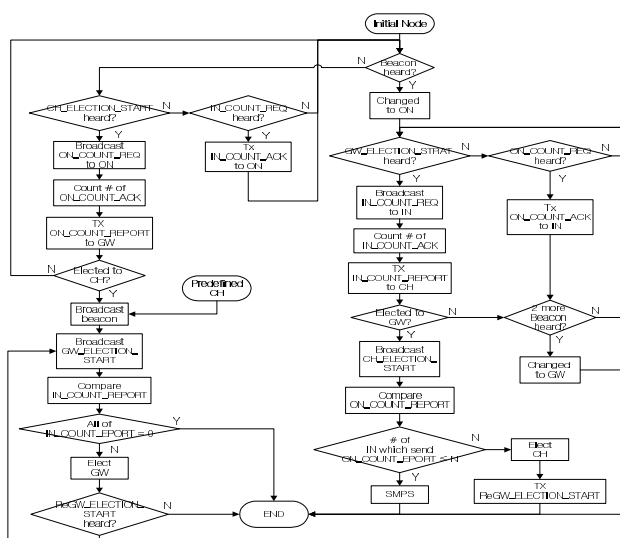


Fig. 2. Flow Chart of Neighbor Node Discovery Algorithm

### 3.2.1 Initial Cluster Set-Up Procedure

One of the CHs is designed as an originating CH, from which other clusters are progressively formed. It broadcasts an advertisement message within its radio range and an IN which has received it sends a joining request message to the CH (refer to Fig. 3). The CH stores the identity of the IN which has sent a joining request message and registers it as a cluster member. The originating CH broadcasts a beacon signal within its radio range and the IN which has heard the beacon becomes an ON.

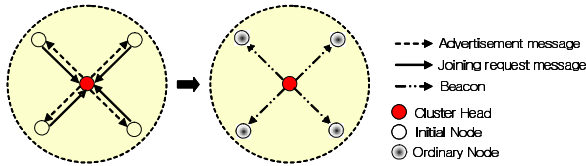


Fig. 3. Initial cluster set-up procedure

### 3.2.2 Gateway Election Procedure

In this procedure, the farthest ON from a CH is elected as a GW among one-hop neighbors of a CH, which allows for maximizing the physical distance between a CH and a GW. To realize this particular procedure, we assume that INs are uniformly distributed, which leads to an acceptable observation that “The farthest ON from a CH will discover the largest number of INs.” Each ON among one-hop neighbors of a CH counts the number of INs within its radio range and then, reports it to the CH. The ON which discovers the largest number of INs is elected as a GW by the CH. Fig. 4 illustrates a simple example, in which Node A includes three INs while node B includes two INs within its radio range. According to the above conclusion, it is desirable that node A is elected as a GW. A contention-based channel access scheme will be used during the gateway election procedure to avoid the collisions between nodes.

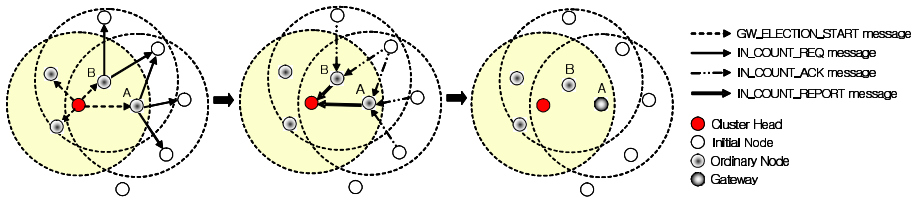


Fig. 4. Gateway election procedure

A CH broadcasts a GW\_ELECTION\_START message within its radio range. An ON which has received this message counts the number of INs within its radio range by the following steps: The ON broadcasts an IN\_COUNT\_REQ message to INs. The IN which has received this message transmits an IN\_COUNT\_ACK message to ON, limiting the transmitted number of this ACK messages to only once even if it has received the multiple IN\_COUNT\_REQ messages from a large number of ONs. The ON reports the received number of ACKs from INs through an IN\_COUNT\_REPORT

message to the CH. The received number of ACKs is equal to the number of INs within the radio range of ON. Analyzing all the received `ON_COUNT_REPORT` messages, CH elects the ON which includes the largest number of INs as a GW. In case of a tie, one with the lowest ID becomes GW. If the values of the received `ON_COUNT_REPORT` messages are all equal to zero, i.e., there is no IN around ONs, the CH does not elect a GW and the clustering algorithm is finished. In case that there is at least one `ON_COUNT_REPORT` message with its value higher than zero, the CH elects a GW and then, proceeds to the cluster head election procedure.

### 3.2.3 Cluster Head Election Procedure

In this procedure, the farthest IN from GW is elected as a CH of the neighbor cluster among one-hop neighbors of GW, which allows for maximizing the physical distance between GW and CH of a neighbor cluster. To realize this particular procedure, we resort to an acceptable observation that “the farthest IN from a GW will discover the least number of ONs.” Each IN among one-hop neighbors of the GW counts the number of ONs within its radio range and reports it to the GW. The IN which discovers the least number of ONs is elected as a CH of neighbor cluster by the GW. Fig. 5 illustrates a simple example, in which Node A includes three ONs while node B includes one ON within its radio range. Following our procedure, the node B is elected as a CH of the neighbor cluster. A contention-based channel access scheme will be used in the course the cluster head election.

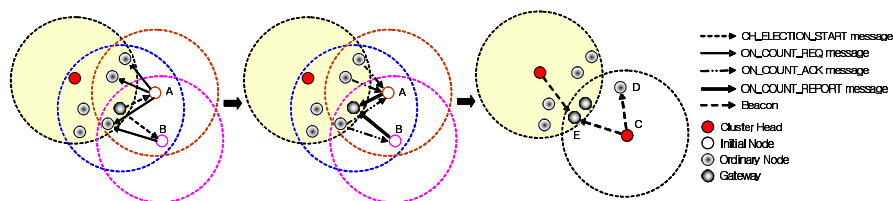


Fig. 5. Cluster head election procedure

A GW broadcasts a `CH_ELECTION_START` message within its radio range. An IN which has received this message counts the number of ONs within its radio range by the following steps: The IN broadcasts an `ON_COUNT_REQ` message to ONs and the ON which has received this message transmits an `ON_COUNT_ACK` message to the IN, limiting the transmitted number of this ACK messages to only once even if it has received the multiple `ON_COUNT_REQ` messages from a large number of INs. The IN reports the received number of ACKs from ONs through the `ON_COUNT_REPORT` message to the GW. The received number of ACKs is equal to the number of ONs within the radio range of IN. Analyzing all the `ON_COUNT_REPORT` messages, GW elects the IN which includes the least number of ONs as a CH of neighbor cluster. In case of a tie, one with the lowest ID becomes CH. If the number of IN which sends the `ON_COUNT_REPORT` message is zero, i.e., there is no IN around GW, the GW does not elect a CH of neighbor cluster and immediately, the clustering algorithm is finished. At least one IN transmits an `ON_COUNT_REPORT` message, the GW elects a CH of neighbor cluster and proceeds to the next gateway election procedure. A newly-elected

CH (Node C) broadcasts an advertisement message within its radio range and the IN which has received it sends a joining request message to a CH. The CH stores the identity of IN which has sent the joining request message and registers it as a cluster member. The CH broadcasts a beacon and an IN (node D) which has heard it becomes an ON. The ON (Node E) which has heard more than two beacons from the different CH becomes a GW.

### 3.2.4 Gateway Re-election Procedure

In order to reduce the overall clustering time throughout the network, it will be useful to perform the clustering process towards every direction in a parallel manner, as illustrated in Fig. 6. A GW which finishes the election of a CH for neighbor cluster (Node B) transmits a GW\_ReELECTION\_START message to a CH (Node A), previously elected itself as a GW. The CH (node A) which has received GW\_ReELECTION\_START message elects another GW out of ONs within its own cluster, following the previous gateway election procedure.

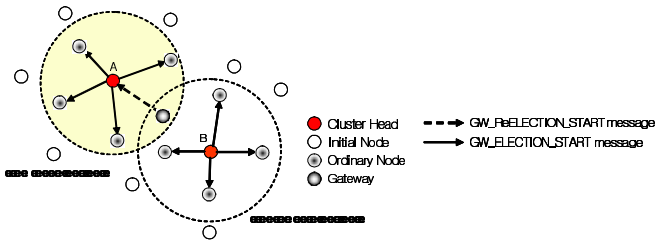


Fig. 6. Gateway re-election procedure

### 3.3 Slave/Master Patching (SMP) Scheme

Most of the clustering algorithms, including the proposed NND algorithm, inherit the problem that the number of GWs is rapidly increasing as a CH is elected although it includes no or a few ONs within its radio range. It is illustrated in Fig. 7. To solve this problem, we consider another feature based on a simple control scheme, named as a Slave/Master Patching (SMP) Scheme. A basic idea behind this scheme is that a CH

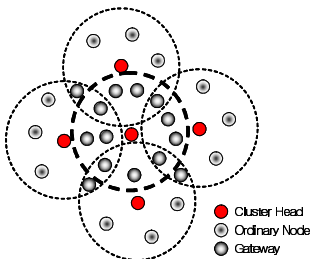


Fig. 7. Undesirable situation of too many GW formed: Illustrative Example

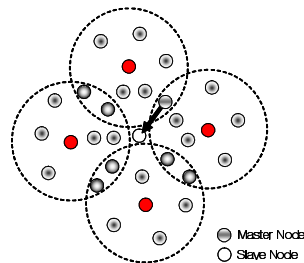


Fig. 8. Example of Slave/Master Patching Scheme

is not elected by GW if the number of INs within a radio range of GW (i.e., the number of received ON\_COUNT\_REPORT message) is lower than a predefined bound, termed as an election threshold  $N$  in the course of our cluster head election procedure ( $N = 0, 1, 2, \dots$ ). To this end, the remaining INs become slave nodes (SNs) of a GW and then, the GW becomes a master node (MN) of the remaining INs, as shown in Fig. 8. The resource and traffic load of the SN can be controlled by MN.

## 4 Performance Analysis

Computer simulation has been performed to evaluate the performance improvement of the proposed NND algorithm over two existing clustering algorithms, Lowest ID algorithm [1] and Topology Discovery algorithm (TopDisc) [3].

The performance measures in our analysis include the numbers of CHs and GWs that have been reduced and the number of ONs that have been increased by the NND algorithm. We assume that sensor nodes are randomly deployed in a field of  $50\text{m} \times 50\text{m}$  as varying the number of sensor nodes from 500 to 1,500. A radio range of each node is fixed to 5m. All results are obtained by taking an average of 500 different runs. Unless stated otherwise, we have chosen the predefined election threshold in the SMP scheme as  $N = 2$ , i.e., GW does not elect CH if the number of INs within its radio range is lower than two.

Fig. 9 through Fig. 11 show the average number of CHs, GWs and ONs formed by each clustering algorithm as the total number of nodes is varied. In Fig. 9, our NND algorithm shows about 6% improvement in the average number of CHs over Lowest ID algorithm. Not much difference from TopDisc algorithm can be found, even if multi-path fading is not taken into account in our simulation. Furthermore, we find the SMP scheme, as a part of our NND algorithm, achieving a significant improvement, as much as 21% over Lowest ID algorithm. Fig. 10 shows that the NND algorithm achieves as much as 13% and 8% decreases in the average number of GWs respectively, as compared with Lowest ID and TopDisc algorithms. Furthermore, the NND algorithm *with* the SMP scheme achieves the average 40 % decrease as compared with Lowest ID algorithm. Meanwhile, Fig. 11 shows that it achieves as much as 29% increase in the average number of ONs as compared with Lowest ID algorithm while achieving the average 11 % increase as compared with TopDisc

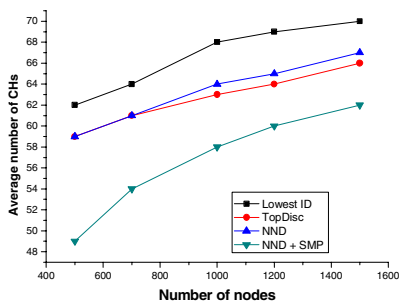


Fig. 9. Average number of CHs

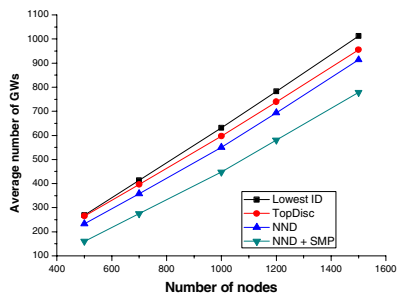


Fig. 10. Average number of GWs

algorithm. Finally, the NND algorithm *with* SMP scheme can achieve as much as 66% increase in the average number of formed ONs as compared with Lowest ID algorithm. In Fig. 12 through Fig. 14, we investigate an effect of the election threshold  $N$  on the performance of our SMP scheme. As  $N$  increases, it shows that the numbers of CHs and GWs are decreasing while the numbers of ONs and SNs are increasing. Important implication is that a mix ratio of all different types of nodes can be controlled by varying the election threshold  $N$  in our approach.

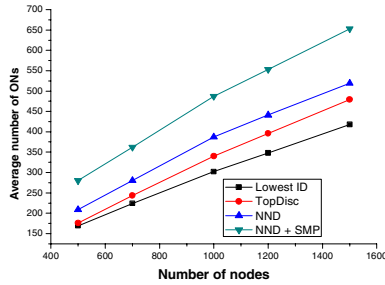


Fig. 11. Average number of ONs

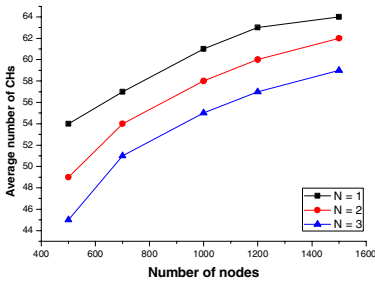


Fig. 12. Average number of CHs

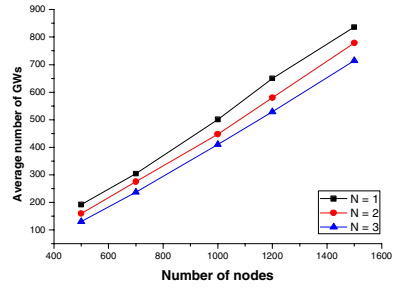


Fig. 13. Average number of GWs

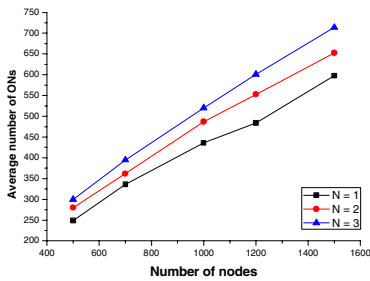


Fig. 14. Average number of ONs

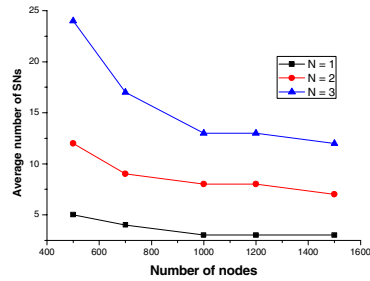


Fig. 15. Average number of SNs

## 5 Conclusion

In this paper, we have proposed a new type of clustering algorithm for a coverage extension, facilitating the energy-efficient medium access and routing controls in the large-scale wireless sensor networks (WSNs). It has been found that the proposed NND algorithm in conjunction with the slave/master patching scheme is effective for implementing the various routing architectures with a single frequency band channel operation. As a mix ratio of all different types of nodes can be controlled by the SMP scheme, it will be a useful and flexible means of deploying the WSNs subject to a limited network lifetime. In particular, the NND algorithm has been successfully employed in our recent development of MAC protocol for WSNs, HiPERMAC (Hierarchically-Paired Evolutionary Radio MAC Protocol) [9], which is designed for implement the hierarchical routing protocol subject to a single frequency channel.

## Acknowledgment

This work has been supported in part by the SAMSUNG Advanced Institute of Technology (SAIT) of Korea and in part by grant No. R01-2003-000-10155-0(2004) from the Basic Research Program of the Korea Science & Engineering Foundation.

## References

1. M, Gerla., J, Tsai.: Multiclustor mobile multimedia radio network. ACM Baltzer Journal of Wireless Networks (1995)
2. D.J, Baker., A, Ephremides.: The Architectural Organization of a Mobile Radio Network via a Distributed Algorithm. IEEE Transactions on Communications, Vol. 29, No. 11 (1981) 1694 - 1701
3. Budhaditya, Deb., Sudeept, Bhatnagar., Badri, Nath.: A Topology Discovery Algorithm for Sensor Networks with Applications to Network Management. Technical Report DCS-TR-441, Department of Computer Science, Rutgers University (2001)
4. B, Krishnamachari.: The impact of data aggregation in wireless sensor networks. in the 22nd International Conference on Distributed Computing Systems Workshops (2002)
5. Zhihui, Chen., Khokhar, A.: Self organization and energy efficient TDMA MAC protocol by wake up for wireless sensor networks. Sensor and Ad Hoc Communications and Networks,2004. IEEE SECON 2004. (2004) 335 - 341
6. T.J, Kwon., Mario, Gerla.: Efficient flooding with Passive Clustering (PC) in ad hoc networks. ACM SIGCOMM Computer Communication Review Volume 32 Issue 1 (2002)
7. Lin, C.R., Gerla, M.: Adaptive clustering for mobile wireless networks. Selected Areas in Communications, IEEE Journal on Volume 15, Issue 7 (1997) 1265 - 1275
8. Chatterjee, M., Sas, S.K., Turgut, D.: An on-demand weighted clustering algorithm (WCA) for ad hoc networks. Global Telecommunications Conference, GLOBECOM '00. IEEE Volume 3 (2000) 1697 - 1701
9. HiPERMAC: Hierarchically-Paired Evolutionary Radio MAC Protocol, Internal Report, Samsung Advanced Institute of Technology, June 2005 (also filed for patent).

# A Short Proxy Signature Scheme: Efficient Authentication in the Ubiquitous World\*

Xinyi Huang<sup>1</sup>, Yi Mu<sup>2</sup>, Willy Susilo<sup>2</sup>, Fangguo Zhang<sup>3</sup>, and Xiaofeng Chen<sup>4</sup>

<sup>1</sup> College of Mathematics and Computer Science,  
Nanjing Normal University, P.R. China  
xinyinjnu@126.com

<sup>2</sup> Centre for Information Security Research,  
School of Information Technology and Computer Science,  
University of Wollongong, Australia  
{wsusilo, ymu}@uow.edu.au

<sup>3</sup> Department of Electronics and Communication Engineering,  
Sun Yat-Sen University, Guangzhou 510275, P.R. China  
isdzhfg@zsu.edu.cn

<sup>4</sup> Department of Computer Science,  
Sun Yat-Sen University, Guangzhou 510275, P.R. China  
isschxf@zsu.edu.cn

**Abstract.** We present a cryptanalysis on the short proxy signature scheme recently proposed in [11] and propose a novel short proxy signature scheme from bilinear pairings. Compared with the existing proxy signature schemes, the signature length of our scheme is the *shortest*. Our short proxy signature scheme satisfies all the properties required for proxy signatures. We prove that our scheme is secure in the random oracle model.

**Keywords:** Proxy Signature, Short Signature, Authentication.

## 1 Introduction

Ubiquitous computing plays an important role in many aspects such as human factors, computer science, engineering, and social sciences. However, Placing computers in human life would face an essential problem, namely, how to implement security and trust among the users that connected to a network. As an example, in a wireless network, the users can connect to a network in anywhere within the broadcast power range. How can they know that they are talking with a real person? Therefore, a necessary authentication scheme must be deployed. In a distributed computing environment, usually, a network is heavily loaded with thousands of users and the bandwidth consumption is a major concern. To

---

\* This work is supported by ARC Discovery Grant DP0557493 and the National Natural Science Foundation of China 60403007.



achieve security without consuming substantial bandwidth is a major challenge to security researchers. In this paper, we will describe an authentication scheme that presents a promise to the minimal use of bandwidth and to providing strong authentication in a “proxy” environment.

The concept of proxy signature can be very useful in cases when a user (say, Alice) wants to delegate her signing right to the other user or proxy (say, Bob). Once the delegation is performed, the proxy can then sign on behalf of the original signer. The notion of proxy signature was introduced by Mambo, Usuda and Okamoto [10]. Based on the delegation type, they classified proxy signatures as full delegation, partial delegation, and delegation by warrant. In the full delegation system, Alice’s private key is given to Bob directly so that Bob can have the same signing capability as Alice. In practice, such schemes are obviously impractical and insecure. In a partial proxy signature scheme, a proxy signer possesses a key, called private proxy key, which is different from Alice’s private key. So, proxy signatures generated by using the proxy key are different from Alice’s signatures. However, in such schemes, the messages a proxy signer can sign is not limited. This weakness is eliminated in delegation by a warrant that specifies what kinds of messages are delegated. Some related works about proxy signatures can be found from [4, 9, 8, 6, 12].

According to whether the original signer knows the proxy private key, proxy signatures can be classified into proxy-unprotected and proxy-protected. In a proxy-protected scheme only the proxy signer can generate proxy signatures, while in a proxy-unprotected scheme either the proxy signer or the original signer can generate proxy signatures since both of them has a knowledge on the proxy private key. In many applications, proxy-protected schemes are required to avoid the potential disputes between the original signer and the proxy signer.

Short signature has attracted a lot of attention since the exploring positive use of bilinear pairing [2]. With bilinear pairings, a digital signature can be as short as 160 bits [3, 1, 5]. Short signatures have a great advantage while the bandwidth of a communication channel is limited. Recently, Okamoto, Inomata and Okamoto [11] proposed a short proxy signature scheme, which allows a much shorter size than other existing schemes. We refer it to as OIO scheme. Unfortunately, we found that the scheme is flawed.

In this paper, we show that their scheme is not secure against a dishonest original signer; namely, given a valid proxy signature, the original signer can forge a valid proxy signature of any new message. We then propose a novel proxy signature, which, we believe, is the shortest proxy signature scheme amongst all existing proxy signature schemes. We also provide a security proof to our novel scheme and show that our scheme is secure against dishonest Alice and Bob and any other polynomial-time adversaries.

The rest of this paper is arranged as follows. In Section 2, we provide the preliminaries of our scheme including bilinear pairings and security assumptions. In Section 3, we give a cryptanalysis on the OIO scheme and show that their scheme is flawed. In Section 4, we describe the model of our proxy signature scheme. In Section 5, we present a novel construction of the shortest proxy

signature. In Section 6, we provide a security proof to our scheme. We show that our scheme is secure against any polynomial adversaries. In the last section, we conclude our paper.

## 2 Preliminaries

### 2.1 Basic Concepts on Bilinear Pairings

Let  $\mathbb{G}_1$  be cyclic additive groups of prime order  $q$  and is generated by  $P$ . Let  $\mathbb{G}_2$  be a cyclic multiplicative group with the same order  $q$ . Let  $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_2$  be a bilinear mapping with the following properties:

1. *Bilinearity*:  $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$  for  $(P, Q) \in \mathbb{G}_1 \times \mathbb{G}_1$  and  $a, b, \in \mathbb{Z}_q$ . Here  $\mathbb{Z}_q$  denotes the definite field of the order  $q$ .
2. *Non-Degeneracy*: There exists  $(P, Q) \in \mathbb{G}_1 \times \mathbb{G}_1$  such that  $\hat{e}(P, Q) \neq 1_{\mathbb{G}_2}$ .
3. *Computability*: There exists an efficient algorithm to compute  $\hat{e}(P, Q)$  for  $(P, Q) \in \mathbb{G}_1 \times \mathbb{G}_1$ .

### 2.2 Security Assumption

#### Definition 1. Computational Diffie-Hellman (CDH) Problem.

Given two randomly chosen  $aP, bP \in (\mathbb{G}_1, +)$  of prime order  $q$ , for unknown  $a, b \in \mathbb{Z}_q$ , compute  $Z = abP$ .

The CDH assumption states that for every probabilistic polynomial-time algorithm  $\mathcal{A}$ ,  $\text{Succ}_{\mathcal{A}, \mathbb{G}_1}^{CDH}$  is negligible.

### 2.3 ZSS Signature Scheme [5]

The ZSS signature scheme proposed in [5] consists of the following algorithms: a parameter generation algorithm  $\text{ParamGen}$ , a key generation algorithm  $\text{KeyGen}$ , a signature generation algorithm  $\text{Sign}$  and a signature verification algorithm  $\text{Ver}$ .

1.  $\text{ParamGen}$ : The system parameters are  $\{\mathbb{G}_1, \mathbb{G}_2, \hat{e}, q, P, H\}$ . Here  $H : \{0, 1\}^* \rightarrow \mathbb{Z}_q$  is a cryptographic hash function
2.  $\text{KeyGen}$ : Randomly selects  $x \in \mathbb{Z}_q^*$ , and computes  $P_{pub} = xP$ . The public key of the signer is  $P_{pub}$ . The secret key is  $x$ .
3.  $\text{Sign}$ : Given a secret key  $x$ , and a message  $m$ , computes  $S = \frac{1}{H(m)+x}P$ . The signature is  $S$ .
4.  $\text{Ver}$ : Given a public key  $P_{pub}$ , a message  $m$ , and a signature  $S$ , verify whether  $\hat{e}(H(m)P + P_{pub}, S) = \hat{e}(P, P)$ .

The security of ZSS signature scheme is based on the security of the  $k$ -CAA problem. For more details, we refer the readers to [5].

#### Definition 2. k-Collusion Attack Algorithm(k-CAA)

For an integer  $k$ , and  $x \in \mathbb{Z}_q$ ,  $P \in G_1$ , given  $(P, Q = xP, h_1, \dots, h_k \in \mathbb{Z}_q, \frac{1}{h_1+x}P, \dots, \frac{1}{h_k+x}P)$ , to compute  $\frac{1}{h+x}P$  for some  $h \notin \{h_1, h_2, \dots, h_k\}$ .

The  $k$ -CAA assumption states that for every probabilistic polynomial-time algorithm  $\mathcal{A}$ ,  $\text{Succ}_{\mathcal{A}, \mathbb{G}_1}^{k\text{-CAA}}$  is negligible. The following theorem has been proved in [5].

**Theorem 1.** *If there exists a  $(t, q_H, q_S, \epsilon)$ -forger  $\mathcal{F}$  using adaptive chosen message attack for the proposed signature scheme, then there exists a  $(t', \epsilon')$ -algorithm  $\mathcal{A}$  solving  $q_S$ -CAA, where  $t' = t, \epsilon' \geq (\frac{q_S}{q_H})^{q_S}$ .*

### 3 An Analysis of the OIO Short Proxy Signature Scheme

Recently, a short proxy signature scheme (OIO for short) was presented in [11]. In this section, we will firstly describe the OIO short proxy signature scheme, then we give an attack to show that the original signer can successfully forge a valid proxy signature of OIO’s scheme.

#### 3.1 Description of OIO Short Proxy Signature Scheme

1. Notations Used in OIO Scheme
  - $\mathcal{O}$ : an original signer;  $\mathcal{P}$ : a proxy signer;  $\mathcal{V}$ : a verifier.
  - $ID_p$ : the ID for a user  $p$ ,  $m_p$ ; a message to be signed by  $\mathcal{P}$ .
  - $\mathcal{H}(\cdot)$ : a hash function  $\mathcal{H} : \{0, 1\}^* \rightarrow \mathbb{Z}_q$ .
2. Key Generation:
  - (a)  $\mathcal{O}$  picks up two elements  $P \in \mathbb{G}_1$  and  $s \in \mathbb{Z}_q$  at random and computes  $V = sP, g = \hat{e}(P, P)$ .  $\mathcal{O}$  sends  $g$  to  $\mathcal{P}$ .
  - (b)  $\mathcal{P}$  picks up a random number  $r \in \mathbb{Z}_q$ , computes  $v_p = g^r$  and sends  $v_p$  to  $\mathcal{O}$ .  $\mathcal{O}$  then computes  $e_p = \mathcal{H}(ID_p, v_p)$  and  $S_p = \frac{1}{s+e_p}P$ .
  - (c)  $\mathcal{O}$  publishes  $g, V, ID_p$  and sends  $S_p$  to  $\mathcal{P}$  using a secure channel.
  - (d)  $\mathcal{P}$  checks whether  $\hat{e}(e_pP + V, S_p) = g$  holds or not.  
As a result,  $\mathcal{O}$ ’s key tuple is {Public-key:  $g, V$ ; Secret-key:  $s$ }.  $\mathcal{P}$ ’s key tuple is {Public-key:  $ID_p, v_p$ ; Secret-key:  $r, S_p$ }.
3. Proxy Signature Generation:  $\mathcal{P}$  computes  $e_m = \mathcal{H}(m_p, v_p)$  and  $Sig_p = (r + e_m)S_p$ . The proxy signature for a message  $m_p$  is  $Sig_p$ .
4. Proxy Verification:  $\mathcal{V}$  first computes  $e_p = \mathcal{H}(ID_p, v_p)$  and  $e_m = \mathcal{H}(m_p, v_p)$ . Then he checks whether  $\hat{e}(e_pP + V, Sig_p) = v_p g^{e_m}$  holds or not.

#### 3.2 An Attack Model of OIO Short Proxy Signature Scheme

Suppose there is a valid message-signature pair  $(m, Sig_m)$ . Since  $Sig_m$  is a valid proxy signature on the message  $m$ , we have  $Sig_m = (r + e_m)S_p$ . Then  $\mathcal{O}$  can compute  $rP = (s + e_p)Sig_m - \mathcal{H}(m, v_p)P$ .

With the knowledge of  $rP$ , the original signer is able to forge a valid signature on any new message. For a new message  $m^*$ ,  $\mathcal{O}$  computes  $Sig_{m^*} = \frac{rP + e_{m^*}P}{s + e_p}$  where  $rP = (s + e_p)Sig_m - \mathcal{H}(m, v_p)P$ ,  $e_{m^*} = \mathcal{H}(m^*, v_p)$  and  $e_p = \mathcal{H}(ID_p, v_p)$  are all known to  $\mathcal{O}$ . We can find it is a valid proxy signature on the message  $m^*$  because  $Sig_{m^*} = \frac{rP + e_{m^*}P}{s + e_p} = (r + e_{m^*})\frac{1}{s + e_p}P = (r + e_{m^*})S_p$  which is indistinguishable to the third party which party ( $\mathcal{P}$  or  $\mathcal{O}$ ) is the signer.

## 4 Outline of Our Short Proxy Signature (SPS) Scheme

Let Alice denote the origin signer and Bob the proxy signer. Our short proxy signature scheme consists of the following algorithms: ParamGen, KeyGen, ProxyKeyGen, ProxySign and ProxyVer.

1. **ParamGen**: Taking as input the system security parameter  $k$ , this algorithm outputs system's parameters:  $\text{Para}$ .
2. **KeyGen**: Taking as input the system security parameter  $k$ , the algorithm generates the secret/public key pair  $(x_i, P_i)$  where  $i \in \{A, B\}$ . That is  $(x_i, P_i) \leftarrow \text{KeyGen}(\text{Para})$ .
3. **ProxyKeyGen**: The original signer Alice and the proxy signer Bob utilize this algorithm to obtain the proxy key which will be used in the ProxySign. That is  $\text{proxykey} \leftarrow \text{ProxyKeyGen}(\text{Para}, x_A, P_A, x_B, P_B, ID_B, m_w)$ .  $m_w$  is the warrant which specifies what kinds of messages are delegated and  $ID_B$  is the identity of the proxy signer Bob.
4. **ProxySign**: The proxy signer utilizes this algorithm to generate the proxy signature. That is  $\sigma \leftarrow \text{ProxySign}(m, \text{proxy key}, \text{Para})$ .
5. **ProxyVer**: Given the public keys of the origin signer and proxy signer, anyone can use this algorithm to check whether a signature is a valid proxy signature. That is  $\{\text{True}, \perp\} \leftarrow \text{ProxyVer}(m, \sigma, P_A, P_B, ID_B, m_w, \text{Para})$

### 4.1 Attack Model

To discuss the Non-Forgeability of our short proxy signature scheme, we divide the adversaries into the following three types:

1. **Type I**: The adversary only has the public keys of Alice and Bob.
2. **Type II**: The adversary has the public keys of Alice and Bob and also has the secret key of Bob.
3. **Type III**: The adversary has the public keys of Alice and Bob and also has the secret key of Alice.

One can find that if our short proxy signature scheme is unforgeable against Type II (or Type III) adversary, our scheme is also unforgeable against Type I adversary.

### Formal Security Notion

#### *Type II Adversary*

We provide a formal definition of existential unforgeability of a short proxy signature scheme under a Type II chosen message attack (*EF-SPS*-adversary). This type of adversaries only has the secret key of the proxy signer and does not obtain the proxy key from the original signer. It is defined using the following game between an adversary  $\mathcal{A}_{II}$  and a challenger  $\mathcal{C}$ .

- **Setup**:  $\mathcal{C}$  runs the algorithm to obtain the secret key and public key pair  $(x_A, P_A), (x_B, P_B)$  representing the keys of the original signer  $A$  and the proxy signer  $B$ , respectively.  $\mathcal{C}$  then sends  $(P_A, P_B, x_B)$  to the adversary  $\mathcal{A}_{II}$ .

- **PublicKey Queries:**  $\mathcal{A}_{II}$  can set the  $i^{th}$  user in the system as the proxy signer. He asks the public key  $P_i$  of the  $i^{th}$  user with the identity  $ID_i$ . In response,  $\mathcal{C}$  generates  $P_i$  and returns  $P_i$  to the adversary  $\mathcal{A}_{II}$ .
- **PSign Queries:**  $\mathcal{A}_{II}$  can request a signature on a message  $m$  with the original signer  $A$  and the proxy signer with the identity  $ID_i$ . In response,  $\mathcal{C}$  outputs a signature  $\sigma$  for a message  $m$ .
- **Output:** Finally,  $\mathcal{A}_{II}$  outputs a target message  $m^* \in \{0, 1\}^*$  and  $\sigma^*$  such that  $\sigma^*$  is a valid proxy signature with the original signer  $A$  and the proxy signer  $B$ .

*Type III Adversary*

We provide a formal definition of existential unforgeability of a short proxy signature scheme under a Type III chosen message attack (*EF-SPS*-adversary). It is defined using the following game between an adversary  $\mathcal{A}_{III}$  and a challenger  $\mathcal{C}$ .

- **Setup:**  $\mathcal{C}$  runs the algorithm to obtain the secret key and public key pair  $(x_A, P_A)$ ,  $(x_B, P_B)$  representing the keys of the original signer  $A$  and the proxy signer  $B$ , respectively.  $\mathcal{C}$  then sends  $(P_A, P_B, x_A)$  to the adversary  $\mathcal{A}_{III}$ .
- **PSign Queries:**  $\mathcal{A}_{III}$  can request a signature on a message  $m$ . In response,  $\mathcal{C}$  outputs a signature  $\sigma$  for a message  $m$ .
- **Output:** Finally,  $\mathcal{A}_{III}$  outputs a target message  $m^* \in \{0, 1\}^*$  where  $m^*$  has never been queried during the PSign Queries and  $\sigma^*$  is a valid proxy signature with the original signer  $A$  and the proxy signer  $B$ .

**Definition 3.** *A short proxy signature scheme is existential unforgeable against chosen-message attacks iff it is secure against both type II and type III adversaries.*

## 5 Our Scheme

1. **ParamGen:** Taking as input the system security parameter  $k$ , this algorithm outputs  $\{\mathbb{G}_1, \mathbb{G}_2, q, \hat{e}, P\}$ , including a cyclic additive group  $\mathbb{G}_1$  of order  $q$ , a multiplicative group  $\mathbb{G}_2$  of order  $q$ , a bilinear map  $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$  and a generator  $P$  of  $\mathbb{G}_1$ . This algorithm also outputs two cryptographic hash functions  $H_0$  and  $H_1$  where  $H_0 : \{0, 1\}^* \rightarrow \mathbb{G}_1$  and  $H_1 : \{0, 1\}^* \rightarrow \mathbb{Z}_q^*$ . We denote the set  $\mathbb{Z}_q^* = \mathbb{Z}_q \setminus \{0\}$  where 0 is the zero element of the field  $\mathbb{Z}_q$ .
2. **KeyGen:** The algorithm generates the original signer Alice’s secret/public key pair  $(x_A, P_A = x_AP)$  and the proxy signer Bob’s secret/public key pair  $(x_B, P_B = x_BP)$ .
3. **ProxyKeyGen:**
  - (a) Alice computes  $D_{AB} = x_AQ_B$ . Here,  $Q_B = H_0(ID_B, P_B, m_w)$ .  $ID_B$  is the identity of the proxy signer Bob,  $P_B$  is the public key of Bob, and  $m_w$  is the warrant. Alice then sends  $D_{AB}$  to Bob.
  - (b) Bob verifies whether  $\hat{e}(D_{AB}, P) = \hat{e}(Q_B, P_A)$ .  
As a result, Bob obtains his proxy key  $(x_B, D_{AB})$ .

4. **ProxySign**: For a message  $m$ , Bob computes  $\sigma = \frac{1}{H_1(m)+x_B}D_{AB}$ . The proxy signature on the message  $m$  is  $\sigma$ .
5. **ProxyVer**: To check whether  $\sigma$  is a valid proxy signature, any one can check:  $\hat{e}(\sigma, H_1(m)P + P_B) \stackrel{?}{=} \hat{e}(Q_B, P_A)$ . If the equation holds, the receiver accepts it as a valid proxy signature; otherwise, rejects it.

The correctness of the scheme can be verified:

$$\begin{aligned} \hat{e}(\sigma, H_1(m)P + P_B) &= \hat{e}\left(\frac{1}{H_1(m) + x_B}D_{AB}, H_1(m)P + P_B\right) \\ &= \hat{e}\left(\frac{1}{H_1(m) + x_B}D_{AB}, (H_1(m) + x_B)P\right) \\ &= \hat{e}(D_{AB}, P) = \hat{e}(Q_B, P_A) \end{aligned}$$

## 6 Security Analysis

### 6.1 Unforgeable Against Type II Adversary

**Theorem 2.** *Let  $\mathcal{A}_{II}$  be a type II adversary who can get a valid signature of our scheme with success probability  $Succ_{\mathcal{A}_{II},SPS}^{EF-CMA}$ . In some polynomial time  $t$ , he can ask  $q_H$  hash queries to the hash function  $H_1$  and  $q_S$  sign queries and  $q_V$  verify queries, then there exists  $\mathcal{B}$  who can use  $\mathcal{A}_{II}$  to solve an instance of CDH problem with the success probability*

$$Succ_{\mathcal{B},\mathbb{G}_1}^{CDH} = Succ_{\mathcal{A}_{II},SPS}^{EF-CMA}$$

*in the same polynomial time  $t$ .*

*Proof.* Given  $P_1 = aP, P_2 = bP$  for some unknown  $a, b \in \mathbb{Z}_q^*$ , we will show how  $\mathcal{B}$  can use the type II adversary  $\mathcal{A}_{II}$  to get the value  $abP$ . Let's recall the definition of the type II adversary  $\mathcal{A}_{II}$ . This type of adversary  $\mathcal{A}_{II}$  only has the secret key of the proxy signer Bob.

$\mathcal{B}$  chooses  $c \in_R \mathbb{Z}_q^*$  and sets the original signer's public key  $P_A = P_1 = aP$ , the proxy signer's public key  $P_B = cP$  and  $Q_B = P_2 = bP$ .  $\mathcal{B}$  returns  $(P, P_A, P_B, Q_B, c)$  to the Type II adversary  $\mathcal{A}_{II}$ .  $\mathcal{A}_{II}$  can ask most  $q_H$  PHash Queries and  $q_S$  PSign Queries to the PHash Oracle and PSign Oracle respectively.  $\mathcal{A}_{II}$  can additionally request the PublicKey Oracle of the other proxy signer's public key he is interested in.  $\mathcal{B}$  will act all these oracles in our proof. After all the queries,  $\mathcal{A}_{II}$  will output a valid proxy signature  $(m^*, \sigma^*)$  such that  $\hat{e}(\sigma^*, H_1(m^*)P + P_B) = \hat{e}(Q_B, P_A)$ . Here, we assume that  $m^*$  has been queried by  $\mathcal{A}_{II}$  to the PHash Oracles before he outputs the signature  $\sigma^*$  of the message  $m^*$ .

In the proof  $\mathcal{B}$  maintains a list, *H-List*, to record all the PHash Queries and the corresponding answers.  $\mathcal{B}$  also maintains another list *PK-List* to record the public key queries and the corresponding answers. We assume that before  $\mathcal{A}_{II}$  asks the PSign Queries with the  $i^{th}$  user is proxy signer,  $\mathcal{A}_{II}$  has obtained the public key  $P_i$  and  $Q_i$  of the  $i^{th}$  proxy signer from the PublicKey Oracle.

1. **Public Key Queries:** In this process,  $\mathcal{A}_{II}$  can ask the  $P_i$  and  $Q_i$  of the  $i^{th}$  proxy signer with the identity  $ID_i$ . For each request,  $\mathcal{B}$  chooses  $x_i, y_i \in \mathbb{Z}_q^*$ , and sets  $P_i = x_iP, Q_i = y_iP$ .  $\mathcal{B}$  then adds  $(ID_i, x_i, y_i)$  to the *PK-List* and returns  $(P_i, Q_i)$  to  $\mathcal{A}_{II}$ .
2. **PHash Queries:** In this process,  $\mathcal{A}_{II}$  can ask at most  $q_H$  PHash Queries. For each request  $m_i$ ,  $\mathcal{B}$  first checks the *H-List*:
  - (a) If there is an item  $(m_j, h_j)$  in the *H-List* such that  $m_j = m_i$ ,  $\mathcal{B}$  sets  $H_1(m_i) = h_j$  and returns  $h_j$  as the hash value of  $m_i$  to  $\mathcal{A}_{II}$ .
  - (b) Otherwise,  $m_i$  has not been requested to the hash oracle.  $\mathcal{B}$  chooses  $h_i \in \mathbb{Z}_q^*$  such that there is no item  $(\cdot, h_i)$  in the *H-List*.  $\mathcal{B}$  then adds  $(m_i, h_i)$  into the *H-List* and returns  $h_i$  to  $\mathcal{A}_{II}$ .
3. **PSign Queries:** In this process,  $\mathcal{A}_{II}$  can ask at most  $q_S$  PSign Queries. For each request  $(m_i, ID_k)$  chosen by  $\mathcal{A}_{II}$ ,  $\mathcal{B}$  first checks the *H-List*:
  - (a) If there is an item  $(m_j, h_j)$  in the *H-List* such that  $m_j = m_i$ ,  $\mathcal{B}$  obtains  $H_1(m_i) = h_j$ .
  - (b) Otherwise,  $m_i$  has not been requested to the hash oracle.  $\mathcal{B}$  chooses  $h_i \in \mathbb{Z}_q^*$  such that there is no item  $(\cdot, h_i)$  in the *H-List*.  $\mathcal{B}$  then adds  $(m_i, h_i)$  into the *H-List* and sets  $H_1(m_i) = h_i$ .

After the check of *H-List*,  $\mathcal{B}$  returns  $\sigma_i = \frac{1}{h_i + x_k} y_k P_A$  to  $\mathcal{A}_{II}$  as the signature of  $m_i$  under the original signer  $A$  and the  $k^{th}$  proxy signer.

After all the queries,  $\mathcal{A}_{II}$  outputs  $(m^*, \sigma^*)$  such that  $\hat{e}(\sigma^*, H_1(m^*)P + P_B^*) = \hat{e}(Q_B, P_A)$ . That is  $\sigma^* = \frac{1}{H_1(m^*)+c} abP$ . Therefore,  $\mathcal{B}$  computes  $(H_1(m^*)+c)\sigma^* = (H_1(m^*)+c) \frac{1}{H_1(m^*)+c} abP = abP$ . Therefore  $\mathcal{B}$  can also solve an instance of CDH problem with the probability  $Succ_{\mathcal{B}, \mathbb{G}_1}^{CDH} = Succ_{\mathcal{A}_{II}, SPS}^{EF-CMA}$ . ■

### 6.2 Unforgeable Against Type III Adversary

**Theorem 3.** *Let  $\mathcal{A}_{III}$  be a type III adversary who can get a valid signature of our short proxy signature (SPS) scheme with probability  $Succ_{\mathcal{A}_{III}, SPS}^{EF-CMA}$ . In polynomial time  $t$  he can ask  $q_H$  hash queries to the hash function  $H_1$  and  $q_S$  sign queries and  $q_V$  verify queries, then there exists another adversary  $\mathcal{B}$  also uses  $\mathcal{A}_{III}$  to obtain a valid signature of ZSS signature scheme [5] with the success probability*

$$Succ_{\mathcal{B}, ZSS}^{EF-CMA} = Succ_{\mathcal{A}_{III}, SPS}^{EF-CMA}$$

*in the same polynomial time  $t$ .*

*Proof.* There two adversaries,  $\mathcal{A}_{III}$  and  $\mathcal{B}$  in our proof.  $\mathcal{A}_{III}$  is the Type III attacker of our proposed short proxy signature (SPS) scheme and  $\mathcal{B}$  is the adversary of ZSS signature scheme [5]. We will show that given Bob’s public key  $P_B$ , how  $\mathcal{B}$  can use  $\mathcal{A}_{III}$  to obtain Bob’s valid signature of ZSS scheme in [5]. As presented in [5],  $\mathcal{B}$  can ask Hash Query and Sign Query to his own Hash Oracle and Sign Oracle.

In the proof,  $\mathcal{A}_{III}$  can ask the PHash Query and PSign Query.  $\mathcal{B}$  will act as these three oracles.  $\mathcal{B}$  chooses  $a, c \in \mathbb{Z}_q^*$  and sets Alice’s public key  $P_A = aP$  and  $Q_B = cP$ . Then,  $\mathcal{B}$  returns  $P_A, P_B, Q_B, a$  to the adversary  $\mathcal{A}_{III}$ .  $\mathcal{A}_{III}$  can ask the following queries:

1. PHash Queries: In this process,  $\mathcal{A}_{III}$  can ask at most  $q_H$  PHash Queries. For each request  $m_i$ ,  $\mathcal{B}$  submits  $m_i$  to his own Hash Oracle and obtains the result  $h_i$ .  $\mathcal{B}$  also returns  $h_i$  to  $\mathcal{A}$  as the answer.
2. PSign Queries: In this process,  $\mathcal{A}_{III}$  can ask at most  $q_S$  PSign Queries. For each request  $m_i$ ,  $\mathcal{B}$  submits  $m_i$  to the Sign Oracle and obtains the result  $\hat{\sigma}_i$ . Then  $\mathcal{B}$  returns  $\sigma_i = ac\hat{\sigma}_i$  to  $\mathcal{A}_{III}$  as the answer. Note that  $\sigma_i$  is a valid proxy signature, this is true because  $\hat{\sigma}_i$  is Bob's valid signature of  $ZSS$ , that is  $\hat{e}(\hat{\sigma}_i, H_1(m_i)P + P_B) = \hat{e}(P, P)$ . Therefore  $\hat{e}(\sigma_i, H_1(m_i)P + P_B) = \hat{e}(ac\hat{\sigma}_i, H_1(m_i)P + P_B) = \hat{e}(\hat{\sigma}_i, H_1(m_i)P + P_B)^{ac} = \hat{e}(cP, aP) = \hat{e}(Q_B, P_A)$

After all the queries,  $\mathcal{A}_{III}$  outputs  $(m^*, \sigma^*)$  such that  $m^*$  is not requested in the PSign Queries and  $\hat{e}(\sigma^*, H_1(m^*)P + P_B) = \hat{e}(Q_B, P_A)$ . Then  $\mathcal{B}$  computes  $\hat{\sigma}^* = (ac)^{-1}\sigma^*$  and  $(m^*, \hat{\sigma}^*)$  is Bob's valid signature in the scheme presented in [5]. This is true because:  $\hat{e}(\hat{\sigma}^*, H_1(m^*)P + P_B) = \hat{e}(\sigma^*, H_1(m^*)P + P_B)^{(ac)^{-1}} = \hat{e}(Q_B, P_A)^{(ac)^{-1}} = \hat{e}(cP, aP)^{(ac)^{-1}} = \hat{e}(P, P)$ . That is to say  $\mathcal{B}$  also find a valid signature of  $ZSS$  signature scheme [5] with the probability  $Succ_{\mathcal{B}, ZSS}^{EF-CMA} = Succ_{\mathcal{A}_{III}, SP_S}^{EF-CMA}$  in the same polynomial time  $t$ . ■

## 7 Conclusion

In this paper, firstly we pointed out that the construction of short proxy signature (OIO scheme) in [4] is insecure. We proceed with a formal definition of short proxy signature scheme, together with three types of adversarial model. Finally, we presented an efficient and short proxy signature, which outperforms any existing proxy signature in terms of signature length, and proved that the scheme is secure in the random oracle model.

## Acknowledgement

The authors would like to express their gratitude thanks to the anonymous referees of the 2nd International Symposium on Ubiquitous Intelligence and Smart Worlds (UISW2005) for the suggestions to improve this paper.

## References

1. D. Boneh and X. Boyen. Short signatures without random oracles. In *Advances in Cryptology, Proc. EUROCRYPT 2004*, LNCS 3027, pages 56–73. Springer–Verlag, 2004.
2. D. Boneh and M. Franklin. Identity-based encryption from the Weil pairing. In *Advances in Cryptology, Proc. CRYPTO 2001*, LNCS 2139, pages 213–229. Springer–Verlag, 2001.
3. D. Boneh, B. Lynn, and H. Shacham. Short signatures from the weil pairing. In *Advances in Cryptology–ASIACRYPT 2001*, LNCS 2248, pages 514–532. Springer–Verlag, 2001.



4. A. Boldyreva, A. Palacio and B. Warinschi. Secure Proxy Signature Schemes for Delegation of Signing Rights. Available at <http://eprint.iacr.org/2003/096>
5. F. Zhang, R. Safavi-Naini, and W. Susilo. An efficient signature scheme from bilinear pairings and its applications. In *Public Key Cryptography (PKC'04)*, LNCS 2947, pages 277–290. Springer–Verlag, 2004.
6. J.-Y. Lee, J. H. Cheon and S. Kim. An analysis of proxy signatures: Is a secure channel necessary? In *Topics in Cryptology - CT-RSA 2003*, LNCS 2612, pages. 68–79. Springer–Verlag, 2003.
7. B. Lee, H. Kim and K. Kim. Strong proxy Signature and its applications. In *Proc of SCIS'01*, pages. 603–08. 2001.
8. B. Lee, H. Kim, and K. Kim. Secure mobile agent using strong nondesignated proxy signature. In *Information Security and Privacy (ACISP'01)*, LNCS 2119, pages. 474–486. Springer–Verlag, 2001.
9. S. Kim, S. Park and D. Won. Proxy Signatures, Revisited. In *Information and Communications Security (ICICS'97)*, LNCS 1334, pages. 223–232. Springer–Verlag, 1997.
10. M. Mambo, K. Usuda and E. Okamoto. Proxy signature: Delegation of the power to sign messages. *IEICE Trans. Fundamentals*, Vol. E79-A, No. 9, Sep., pages. 1338–1353, 1996.
11. T. Okamoto, A. Inomata, and E. Okamoto. A proposal of short proxy signature using pairing. In *International Conference on Information Technology (ITCC 2005)*, pages 631–635. IEEE Computer Society, 2005.
12. H.-U. Park and I.-Y. Lee. A digital nominative proxy signature scheme for mobile communications. In *Information and Communications Security (ICICS'01)*, LNCS 2229, pages. 451–455, Springer–Verlag, 2001.

# The Design and Implementation of Secure Event Manager Using SPKI/SDSI Certificate\*

YoungLok Lee<sup>1</sup>, HyungHyo Lee<sup>2</sup>, Seungyong Lee<sup>1</sup>,  
HeeMan Park<sup>1</sup>, and BongNam Noh<sup>1,\*\*</sup>

<sup>1</sup> Dept. of Information Security, Chonnam National University, Gwangju 500-757, Korea  
{dogu, birch, hareup, bongnam}@lsrc.jnu.ac.kr

<sup>2</sup> Div. of Information and EC, Wonkwang University, Iksan 570-749, Korea  
hlee@wonkwang.ac.kr

**Abstract.** In the ubiquitous computing environment new service components should be able to connect to networks at any time, and clients also should be able to use them immediately even without extra settings. Jini is one of the widely used middlewares today. Although event management is an essential component of ubiquitous middlewares, Jini is distributed without event management service. Accordingly, we design and implement the event manager based on Jini and suggest three methods in which only right event consumer can listen to the event using Access-Control Lists and SPKI/SDSI certificates. In the proposed method, our event manager controls the access of events by putting trust checking engine on Jini.

## 1 Introduction

We only assume a dim prospect of what the ubiquitous computing enabled future might perhaps bring and do not clearly know what is coming. For this reason, we do not know in what ways the ubiquitous computing scenario can be abused by ingenious attackers and do not know who the attackers are going to be.

The important thing is to identify which objects exactly we want to protect. The urgent object to be protected is the event, among many of those objects. There are various events ranged from low level signals generated by sensors to deduced valuable information in high level. Users in the ubiquitous computing environment should be able to adapt themselves to their current context information and high level information generated by these events. If user's event information is illegally achieved, someone can illegally generate a dossier that all the event information issued in the supermarkets, airports, bookstores, or banks are merged and also use it without user's acknowledgement. Above all, the event management is important because the illegal modification of generated events results in a wrong adaptation of users who use the event.

---

\* This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment).

\*\* Corresponding author.

Jini, the Home Network middleware helps new service components connect to Home Networks at any time and helps clients promptly use them without extra settings, and even in the case of service component upgrade, the existing client service can operate with no problems. However, there is no event service implemented in Jini. Instead, it provides JavaSpace, storage system based on object and attributes in order to use various service objects.

By modifying the JavaSpace, we develop the event manager to manage events and control the access of events in order that only right users with the authority granted by the event generator can take event.

This paper consists of as follows;

Chapter 2 shows related researches, and chapter 3 explains the design and implementation of event management system based on JavaSpace. Chapter 4 suggests three methods to control the access of events by event manager, using SPKI/SDSI certificates and ACLs(access control lists). Chapter 5, among those three methods, describes the prototype of the second method which is adaptable in actual life and compares other event manager services. Finally, chapter 6 makes a conclusion and further research works.

## 2 Related Researches

Among middlewares of ubiquitous computing environment are Gaia, M3-RTE, Aura[1,2,3], etc. Each has the similar structure in its event system, however, does not include the security.

One of middlewares is the Gaia of the University Illinois under active research. The event manager of Gaia[4] satisfies many of general needs in event management. The event manager distributes load among multiple machines. It creates event channel factories remotely on pre-configured machines whenever it decides that the existing event channel factories are already overloaded. The event manager is also responsible for recreating those channels if any supplier of events complains that the particular event channel no longer exists. In essence, the event manager keeps state for the channels it creates and recreates them if they crash, transparent to the event consumers. Event manager service implementation of Gaia makes use of CORBA event service as the basic event storage. Nonetheless, Gaia depends on the basic security policy in the event manager as well.

Another is the Context Observer of Carnegie Mellon University. It provides information about the physical context and report events in the physical context back to Prism and the Environment Manager. Prism and the Environment Manager are components of the Aura[3]. Context Observers in each environment may have different degrees of sophistication, depending on the sensors deployed in that environment.

## 3 Event Management System Based on JavaSpace

Jini[5] is a middleware in composing the home networking. The purpose of Jini is to accomplish "Network plug and Work". Although new printer components are connected to network, Jini should be what clients immediately use it without extra setup. Also despite the upgrade of service components, it has no problem in running the existing client with no extra setting.

Although event management is a necessary component of ubiquitous middleware, Jini is now being distributed with no event management service. This section describes how we design and implement the event manager which manages events, using JavaSpace[5].

### 3.1 Process Procedure of Event Service

Our event manager (JS-EM : Event Manager based on JavaSpace) is made by modifying JavaSpace[5]. The procedure that the event consumer takes events generated by the event producer is as follows:

1. As JS-EM itself is registered as service in the Jini LookUp service, the event consumers or the event producers can search the JS-EM and use it. An event consumer registers herself to JS-EM as a listener of the event he or she is interested in.
2. JS-EM takes the stub of event listener for the communication with the event consumer through web server and by doing this, there accomplishes the channel between event listener and JS-EM.
3. The event producers write its events to the JS-EM.
4. Events written to the channel in JS-EM are transmitted to the event consumer via previously registered event listener.

Our event manager provides a model for decoupled communication among event consumers and event producers. It allows creating and deleting channels. And through our event manager service, the context management service is able to generate the high level context information.

### 3.2 Modified JavaSpace

JavaSpace is found through lookup service as similar to other services of Jini and used through Proxy. JavaSpace is in charge of saving objects. In order to use JavaSpace as event manager, some of problems should be solved in advance. If the event producer writes the event to JavaSpace through write( ) operation, one of APIs of JavaSpace, the event consumer takes the event through read( ) operation.

Because, however, this JavaSpace of the general structure does not fully play a role as event manager, we add necessary interfaces into JavaSpace and modify some Classes of JavaSpace. Our event manager made by using the modified JavaSpace will be reused with the enlarged function when we implement Prototype of our Event Manager in Chapter 5.

## 4 Methods of Secure Event Management Using SPKI/SDSI

Ubiquitous services must appropriately adapt to the context information of the user. In need of privacy protection and proper adaptation, context information should be generated from the accurate event information and only right possessor of the authority of the event should utilize it.

The ad hoc network environment introduces fundamental new problems. One is the absence of an online server, and another is secure transient association. Authentica-

tion is one of the most interesting security problems in ad hoc networking, because much of the conventional wisdom from distributed systems does not quite carry over. For solving these problems, we use SPKI/SDSI(Simple Public Key Infrastructure/Simple Distributed Security Infrastructure) certificates.

This chapter, after summarizing the name certificate and authorization certificate structure and certificate chain discovery algorithm, describes three methods suggested by us in order to control the access to the event.

## 4.1 SPKI/SDSI Name Certificate and Authorization Certificate

### 4.1.1 Name Certificate

SPKI/SDSI Name certificate is binding between subject and local name. Local name is defined at any rate based on public key of issuer. Name certificate consists of 4-tuple[6].

<Issuer, Local Name, Subject, Validity>

The principals are public-key in SDSI. Issuers sign certificate with his or her private key. Local name, the one that issuer hopes to bind with the subject, consists of public key of issuer and more than one principal. Name in SDSI is defined only locally. The issuer controls the name space.

Subject is a principal or a name, which is the target bound to the Local Name and simultaneously receiving the certificate. If subject has a name, then that name has no global meaning but is defined only by the principal whose name space it is in. The name can only be bound to a key by a name certificate issued by the principal controlling the name space. What is assigned to subject is public key or local name composed of more than one public key or more than one local name. Validity is the period during which this certificate is valid.

### 4.1.2 Authorization Certificate

Authorization certificate is the one that certificate issuer gives some other subject the right to access a resource, such as reading a file. Authorization certificate is the same with the name certificate excluding the authorization-tag which grants authority to the subject and it has delegation bit. Authorization certificate consists of 5-tuple as follows[7]. This certificate can be combined with other certificates to create a chain of authorization, and it can be verified when accompanied by a valid signature.

<Issuer, Subject, Delegation bit, Authorization-tag, Validity>

### 4.1.3 Certificate Chain Discovery in SPKI/SDSI

Certificate Chain Discovery Algorithm"[8] is the one that searches, in his certificate cash, the name certificates and authorization certificates related to the subject in ACL transmitted from the server. Generally, clients run this algorithm.

The inputs of this algorithm are ACL(access-control list) for a protected resource and a collection of SPKI/SDSI certificates that client store in her or his certificate cash. This algorithm determines whether a given principal or set of principals, represented by their public keys, is authorized to access the protected resource.

## 4.2 Secure Methods to Manage Events

This section suggests three methods to manage events and implement the prototype of the second method, by judging that it is appropriate to the current ubiquitous environment among three methods

### 4.2.1 ACL and Encoded Event

When the owner of the event producer first installs a sensor, it provides the sensor with ACLs in secure satisfactory solution – physical contact. The ACL is the same with security policy that an owner of the sensor provides authority to use her event to a principal. As the event producer issues events, it sends encoded events and ACLs to an event manager, and the events are saved in the channel matching event type. Since then, the event producer sends only the encoded event objects generated into the related channel, managed by event manager.

After the event consumer registers herself to the event channel, the event manager sends all listeners of the event the encoded event and the ACLs related to the event. Accordingly, the event consumer able to decode the ACLs can also decode the encoded event and takes an action for adaptation.

### 4.2.2 SPKI/SDSI Certificate Manager in Middleware

Clients in ubiquitous computing environment are gadgets of micro-sized and low battery and most of them with no computing power. Consequently, gadgets have limit on computing to check trust relations.

Our second method to securely manage events is that only the event consumer who has proper right for any specific event can have the right to register herself as the listener of that event to event manager. Accordingly, in order to register oneself as the event listener, the event consumer requests JS-EM to check whether itself has the right registration authority, or not by sending its possessing SPKI/SDSI name certificates, authorization certificates, and event listener objects. After the checking process is complete, if the event consumer is estimated as the authorized one, JS-EM registers it as the listener of the event, if not, JS-EM destroys the registration.

More details of this method are explained in chapter 5.

### 4.2.3 Event Consumer Verifying Trust Relationship

This method is most suitable for the event consumer in case of having computing ability. This method is similar to 4.2.2, but it is differ that the event consumer checks the trust relationship between subject of ACL and herself. The event consumer registers herself as an event listener on event channel in JS-EM related to the event. At first, the event producer sends ACLs to the channel that JS-EM manages. And after then, the event producer sends only plain event objects to the channel. If the event producer writes the event to the JS-EM, the JS-EM requests to verify the authority by returning the ACL related event channel to the event consumer. The event consumer inputs both of name certificate and authorization in his or her cash and just received ACL in “Certificate Chain Discovery” algorithm so as to verify that it owns the authority to bring out the event. If he finds a certificate chain, he sends the chain to the JS-EM then the JS-EM sends the related event to the event consumer.

## 5 Prototype Implementation and Analysis

### 5.1 System Implementation

Our prototype is implemented in Linux/Windows OS, JDK 1.3 and Jini 1.2 development environment. We implemented our event manager by modifying JavaSpace as described at chapter 3. In this chapter, we explain prototype in order for only right users to take the event. Our prototype checks whether the event consumer is authorized to access the event or not by communicating between event manager and certificate manager. We define and use ACLs and SPKI/SDSI certificates for this trust checking, and extend the event manager we already implemented. We make and add the LRM(listener registration manager) and SSCM(SPKI/SDSI certificate manager) into Jini.

ACL involves security policy, which the event producer delegates authority to the event consumers for receiving the event. When an event sensor is installed, owner of the sensor provides her sensor with ACLs describing event authorization policy. Since then, the owner of sensor can modify or add another ACLs if necessary. A JS-EM is registered as services in the Jini LookUp service, and the event consumers or the event producers search the JS-EM. Now they are able to use it.

The procedure that only the right event consumer takes the event generated by the event producer is as figure 1.

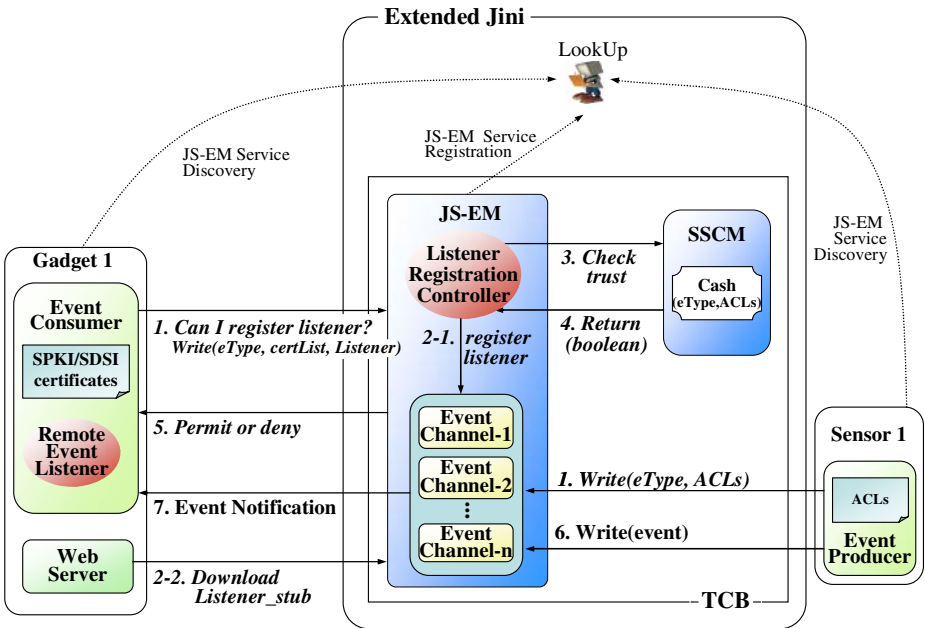


Fig. 1. Secure event management scenario in extended Jini

1. When installing, the event producer, sensor1, sends both ACLs and event type that she owns to the SSCM. Speaking in more detail, when sensor1 boots, sensor1 sends the trust verification information, a type of class objects containing ACLs and event types, to JS-EM. As SSCM was already registered as a listener to receive the trust verification information, JS-EM just notifies this information to the SSCM. Since JS-EM and SSCM are in TCB(trusted computing base) relationship, SSCM securely stores pair of the event type and ACLs into its cash. On the other hand if an event consumer, gadget1, wants to listen to event type E1, it will send a chain of its certificate list, an event type E1, and own listener for that event to JS-EM.
2. LRC of JS-EM takes a role of registering listener of event type for the interim period and locks event objects of the event type not to be taken away for that time. JS-EM takes the stub of the event listener for the communication with the event consumer through web server and by doing this, they accomplish the channel between event listener and JS-EM. And then, LRC relays the event type and certificates to SSCM within TCB and requests to check whether it has the proper authority.
3. SSCM checks the authority of the event consumer by using the certificate chain discovery algorithm.
4. SSCM returns a boolean value as the result to the event consumer
5. If the returned Boolean value from the SSCM is true, LRC remains the listener as it is and sends a message of registration permission to the event consumer. If false, it sends a message saying “denied” and annul the registered listener.
6. Now sensor 1 issues an event object and writes it on JS-EM.
7. JS-EM notifies the events to the listeners of the event objects written by 6.

## 5.2 Structure of the Components in Our Prototype

When installing sensor, the owner of the sensor or the domain administrator saves ACLs by using secure ways, such as physical contact. When a sensor is activated, it takes procedures to send own ACLs and event type to SSCM via the event manager. The simple explanation of the components consisting of our prototype is as follows.

### 5.2.1 ACLs

The structure of ACLs also is composed of 5-tuple like authorization certificates. The example specified below is our ACL's structure.

$$\langle K_{\text{domain\_admin}}, K_{\text{gadget1}}, 0, \text{“permit”}, ((05-06-01, 05-08-20) \wedge (13:50:10, 16:20:35)) \rangle$$

This ACL means that gadget1 can take the sensor's events during the valid expiration date. To send this ACL to SSCM, the event producers use the API function write( ) of JS-EM.

$$\text{JS-EM.write( eventType, ACL-List, LeaseTime)}$$

### 5.2.2 SPKI/SDSI Certificate Manager.

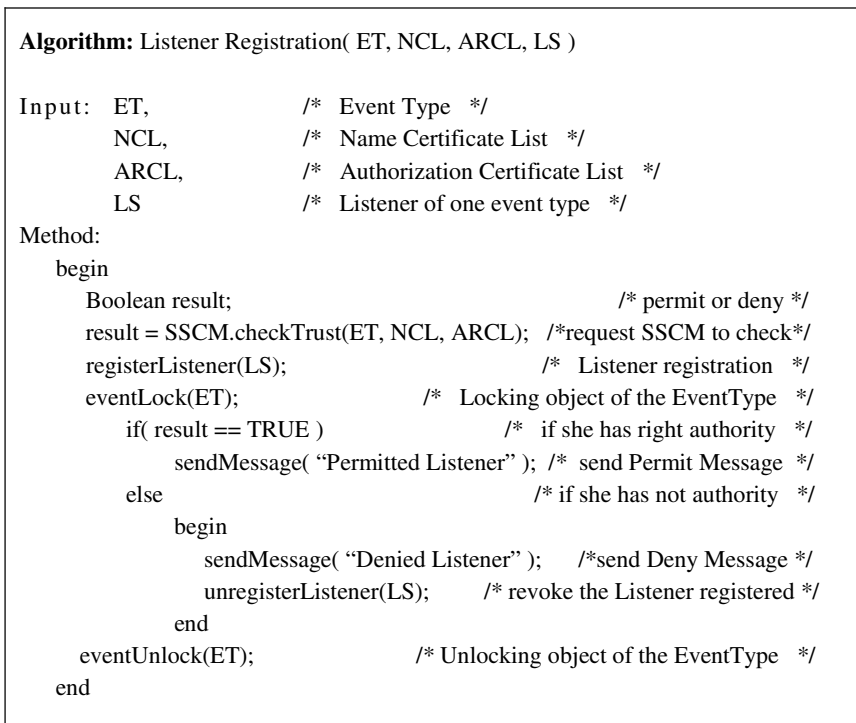
SSCM(SPKI/SDSI Certificate Manager) scrutinizes whether legitimate authorization link ranged from a principal of SPKI/SDSI certificates presented by the event consumer to the subject in ACLs, connecting to the event type stored in own cache is



existed. This test is achieved by using the certificate chain discovery algorithm. Then, SSCM returns a boolean value to the LRC.

### 5.2.3 Listener Registration Controller

LRC(listener registration controller) is needed to register only the listeners who have the authorized rights, so that only registered event consumers can listen to the events for given specific event types. Once LRC receives the event type, SPKI/SDSI certificates, and listener from a event consumer, it registers the event listener and requests the SSCM to check whether the event listener can take event objects of the event type or not. If LRC receives a true value, it keeps the listener registered. If not, it returns denial message to them and destroys the listener. The algorithm LRC performs is as Figure 2.



**Fig. 2.** Listener Registration Algorithm

## 5.3 System Analysis

While Gaia event manager service uses basic event storage of CORBA, our JS-EM uses the modified JavaSpace. Gaia has not controlled the access to events. However, our service can do it by using ACLs and SPKI/SDSI certificates. Event Type is fixed in Gaia, but it is not fixed in our event manager. Both Gaia and ours allow multi event channels and the fact that those event channels are able to be distributed where multi-computing machines are common.

The table 1 is the comparison of our event manager service and other event service.

**Table 1.** The comparison of our event service and Gaia

	Gaia's event service	Our JS-EM
Basic event storage	CORBA	modified JavaSpace
Event access control	Not yet	Yes
Multi event channel	Supported	Supported
Event Type	Fixed	Unfixed
Loading channel	On multi machine	On multi machine

## 6 Conclusions and Further Work

By using JavaSpace, we implement event management system among middleware components suitable for ubiquitous computing. We also suggest three methods in order to manage events safely, and design and implement the second one among those three, because now there is a computing power limitation to gadgets or sensors.

For the purpose of controlling events, we expanded JINI by adding event manager and certificate manager, which are not existed in JINI. This is to provide events only to the event consumer with right authority by interactions between event manager and certificate manager. This paper focuses on where the streamline of event should be controlled rather than the performance of event manager

The common language for expressing security policy such as ACLs is important because it can be moved through networks. In order to solve this problem, policy-decision-point(PDP) module can be implemented by using XML. We also will implement PDP in SSCM who understands XACML(extensible access control markup language) rule context for event producers.

## References

1. Manuel Román, Christopher K. Hess, Renato Cerqueira, Anand Ranganathan, Roy H. Campbell, and Klara Nahrstedt: Gaia: A Middleware Infrastructure to Enable Active Spaces. In *IEEE Pervasive Computing*( Oct-Dec 2002 ) 74-83
2. A. Rakotonirainy, J.Indulska, W.W Loke, A.Aaslavsky: Middleware for Reactive Components: An Integrated Use of Context, Roles, and Event Based Coordination, *Lecture Notes In Computer Science*, Vol. 2218 . Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms, Heidelberg( 2001 ) 77-98
3. J. P. Sousa and D. Garlan: Aura:an architectural framework for user mobility in ubiquitous computing environments. In Proceedings of the 3rd Working IEEE/IFIP Conference on Software Architecture, Montreal, Canada( August 2000 )
4. B.Borthakur: Distributed and Persistend Event System For Active Spaces. In: Master Thesis in Computer Science, Urbana-Champaign: University of Illinois at Urbana-Champaign( 2002 )
5. Philip Bishop and Nigel Warren: *JavaSpaces IN PRACTICE*, Addison-Wesley( 2003 )
6. Andrew J. Maywah: An Implementation of a Secure Web Clent Using SPKI/SDSI Certificates. Master of thesis, M.I.T, EECS( May 2000 )
7. Carl M. Ellison, Bill Frantz, Butler Lampson, Ron Rivest, Brian M. Thomas, Taut Ylonen: SPKI Certificate Theory. RFC2693( September 1999 ).
8. Dwaine Clarke, Jean-Emile Elien, Carl Ellison, Matt Fredette, Alexander Morcos, and Ronald L. Rivest: Certificate Chain Discovery in SPKI/SDSI. *Journal of Computer Security*. volume 9, Issue 4( December 2000 ) 285-322

# Object Reminder and Safety Alarm

Chi-yau Lin, Chia-nan Ke, Shao-you Cheng, Jane Yung-jen Hsu, and Hao-hua Chu

Computer Science and Information Engineering,  
National Taiwan University, Taiwan

{r93922129, r93922109, r93922070, yjhsu, hchu}@ntu.edu.tw

**Abstract.** This paper introduces a novel approach to enhancing safety through RFID technology, location tracking, and monitoring person-object interaction. We design and develop RFID-based wearable devices for (1)tracking people's locations, (2)monitoring person-object interactions, and (3)tracking objects' locations. An intelligent object reminder and safety alert system is proposed to relief the common safety-related worries many of us face in our everyday lives - "Where did I leave my keys?", "Did I turn off the stove?", or "Did I close all the windows in my house?" etc. Experimental results on the precision of object identification and location tracking are also presented.

## 1 Introduction

This research aims to design an intelligent alert system by tracking the behavior of users living within the space. By attaching cheap, readily available, passive RFID tags on everyday objects, such as wallets, key chains, briefcases, shoes, spoons, trash cans, refrigerators, paintings, floors and doors, etc. and embedding tiny, mobile RFID readers on wearable personal items such as rings or a watches, the smart environment can unobtrusively monitor human interactions with these RFID-tagged objects in the physical space. By monitoring such person-object interactions and tracking people's indoor locations, we can infer high-level safety-related activity context, such as a person who turns on a stove in the kitchen (touching the stove knob) and then walks away for a long time (he/she may forget to turn off the stove), a person who opens a window and then leaves home (he or she may be in a risk of burglary), a left-along toddler who holds small items (such as coins, buttons, marbles, beads) and is in high risk of choking on them, etc.

In Section 2, we start by describing the hardware devices and infrastructure. Section 3 presents the proposed multiagent system architecture. Experimental results are given in Section 4, followed by related work and conclusion in Sections 5 and 6.

## 2 Hardware Devices

In the proposed system, we attach short-range RFID tags to everyday objects for identification. Meanwhile, RFID readers are embedded in wearable personal items, such as finger rings or wrist watches, to identify tagged objects that are being handled by the user.

For location tracking, a user carries a PDA equipped with the Ekahau Positioning Engine 2.1 (EPE). It utilizes existing Wi-Fi network infrastructure to facilitate user mobility and asset visibility.

### 3 System Architecture

Two kinds of sensed data are used in our system. As Figure 1 depicts, one sort is RFID-tagged object and the other one is Wi-Fi signal via WLAN network. Our system is a multi-agent based system equipped with wearable RFID reader and Ekahau Wi-Fi software positioning engine. Besides, three agents are built to manage sensed data and cooperate in our system.

The first is Object Tracker Agent. From our wearable RFID-Tag reader, it can read the tag id embedded on the surface of objects. Furthermore, it can identify the object from its tag id, and other properties such as category, owner and RFID information that the object has from its object domain mapping.

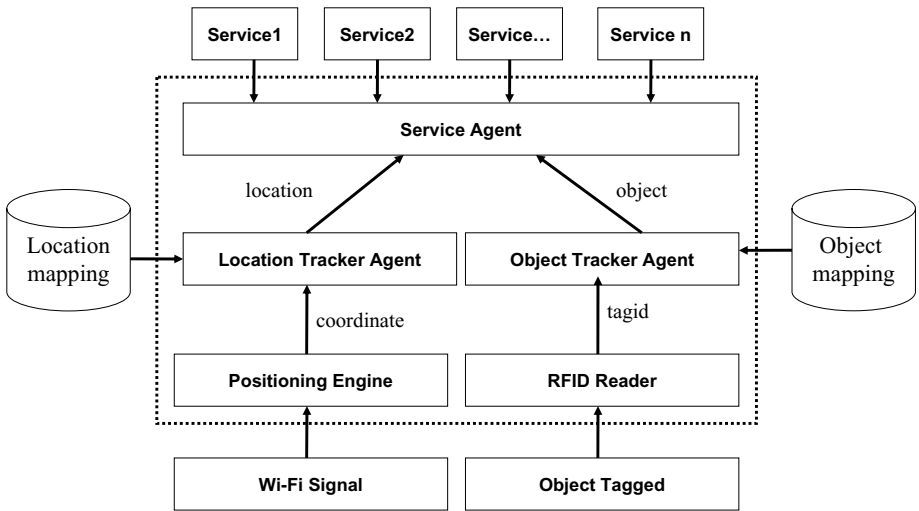


Fig. 1. System architecture based on multi-agents system

The Location Tracker Agent can receive the data transmitted from Ekahau Wi-Fi positioning engine. The Ekahau positioning system returns the coordinates of people’s location. The Location Tracker Agent translates the coordinates into places that people can understand.

The last agent is Service Agent. It can supply services to people. The Service Agent can communicate with Object Tracker Agent and Location Tracker Agent, while people want certain service. The details are presented in section 3.3.

Finally, we will describe how the agents communicate, pass message, and parse message to handle the request.

### 3.1 Object Tracker Agent

Our Object Tracker Agent is a reactive agent. Based on the RFID system, it can percept the object when human touched.

**Object Recognition.** The Object Tracker Agent can recognize the object that the person held. And it is able to process the data in logic, because it can use rules to make inference, choose courses of action and answer questions. What is the knowledge base in the Object Tracker Agent? We built an object domain with an explicit description of an object. Besides, Object Tracker Agent not only maintains the data of objects, it can also communicate with other agents through a interaction protocol.

### 3.2 Location Tracker Agent

In EPE 2.1, the location means a coordinate in a plat map with a layer of certain building. But for our Location Tracker Agent, it can recognize the physical location of the person, instead of just a coordinate.

**Location Tracking with Mobile Device.** Location Tracker Agent is built in our notebook which has installed EPE 2.1. And Ekahau Client is installed in the notebook and PDA. EPE 2.1 will monitor the location of Ekahau Clients. If the location changing, Location Tracker Agent captures the current time and stores the location information in its repository. It translates the location information into Area instead of the coordinate return from EPE 2.1. Area means a block like lab room, toilet, hallway, and so on. Figure 2 shows the Area we defined in our experimental environment.

### 3.3 Service Agent

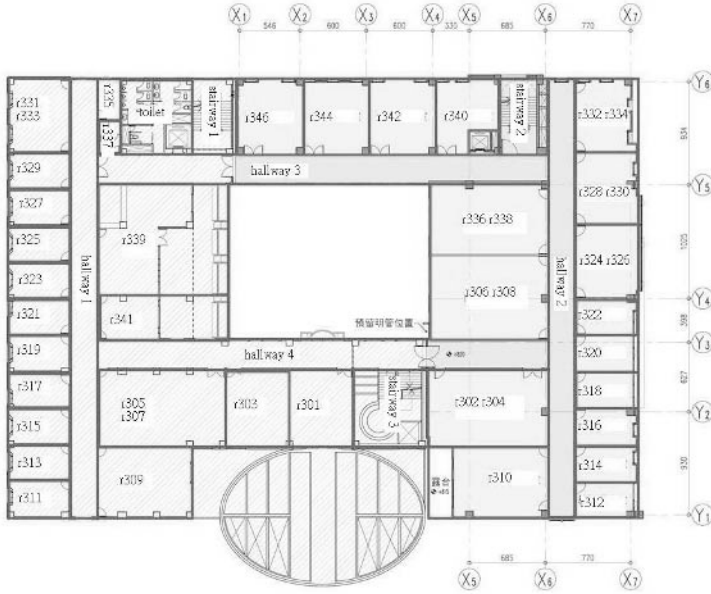
The Service Agent can supply various services while people request. First , we implemented the simple combination of Location Tracker Agent and Object Tracker Agent to find the location of an object. Last, we set rules to detect if any dangerous object ever touched by human, and monitor the interaction between human and objects.

**Object Locating.** Figure 3 describes a simple scenario for our Object Locating Service. The situation occurs to everyone in our daily life. Searching a certain thing is really tedious for people, especially when we need the item in emergency. So how to find the item as soon as possible is a really practical problem. We solved the problem with combination of object and location information.

**Safety Alarm.** In the same scenario, we describe another scenario in Figure 4 to provide our Safety Alarm Service. The service can remind John to avoid an accident due to his carelessness.

**Service Features.** Our services focus on some practical problems and have many features.




First, our smart device is a wearable item that embedded a RFID reader, and a personal server in his pocket. Second, our system is no range limit. Second, our smart



**Fig. 2.** The 3rd floor plan of Computer Science and Information Engineering building at National Taiwan University

	<p>When John came off work, he may put his key chains on the hallway, then go to the living room.</p>
	<p>In the living room, he may put his bag on the sofa, and go to his bedroom.</p>
	<p>He wants to take a shower. So he takes off his clothes and move his wallet from the pocket to his bed. After taking a shower, he goes to the kitchen for cooking.</p>
	<p>After having a dinner, he wants to take a walk outside. So he needs to find out his key chains. At that time, Object Locating Service could help him find out his key chains in a short time. John doesn't need to waste his time and energy searching for each room or place at home.</p>

**Fig. 3.** Object Locating Scenario

	<p>When John is cooking in the kitchen. A “phone ring” event may occur, so he would like to answer the phone in the living room.</p>
	<p>John may talk to his friend for a long time and forget to turn off the gas burner in the kitchen. A fire accident may occur from his carelessness.</p>
	<p>Safety Alarm Service can remind him to turn off the gas burner because the service agent keeps track the object interaction with him and locates his position at home continuously.</p>

**Fig. 4.** Safety Alarm Scenario

device is easy to use, because our smart device is an agent-based system. We design and implement a multiagent system, so our system can update automatically. In other words, our system is less demand for user intervention because of the autonomous of agents, the person will have a clear user interface, and needn't to set any preference in advance.

### 3.4 Agent Communication

The communication of our agents is a FIPA like contract interaction protocol specification. Our communication message format contains the identification of which agent sends a request, and which action that the agent requests another agent to do, and the time when an agent makes a request, and other information needed to pass to another agent. When the agent received a request, it can understand the meaning of the request,

**Table 1.** Communication Messages

#### Messages for Object Locating

Sender	Receiver	Request	Time Stamp	Content
Service Agent	Object Tracker Agent	request object	system time	object name
Object Tracker Agent	Service Agent	—	system time	time or location
Service Agent	Object Tracker Agent	get location	system time	object name
Location Tracker Agent	Service Agent	—	system time	object location

#### Messages for Safety Alarm

Sender	Receiver	Request	Time Stamp	Content
Service Agent	Object Tracker Agent	dangerous object	system time	—
Object Tracker Agent	Search Agent	—	system time	Yes or No
Service Agent	Location Tracker Agent	location changed	system time	—
Location Tracker Agent	Service Agent	—	system time	Yes or No

and response to the request, therefore the agents can negotiate with each other. Table 1 is the message format that we defined for agents.

**Agents in Object Locating Service.** People always forgot the location where they put their object, the service of Object Locating will remind him his missing object. Figure 5(a) shows the detail of interaction among agents in Object Locating Service.

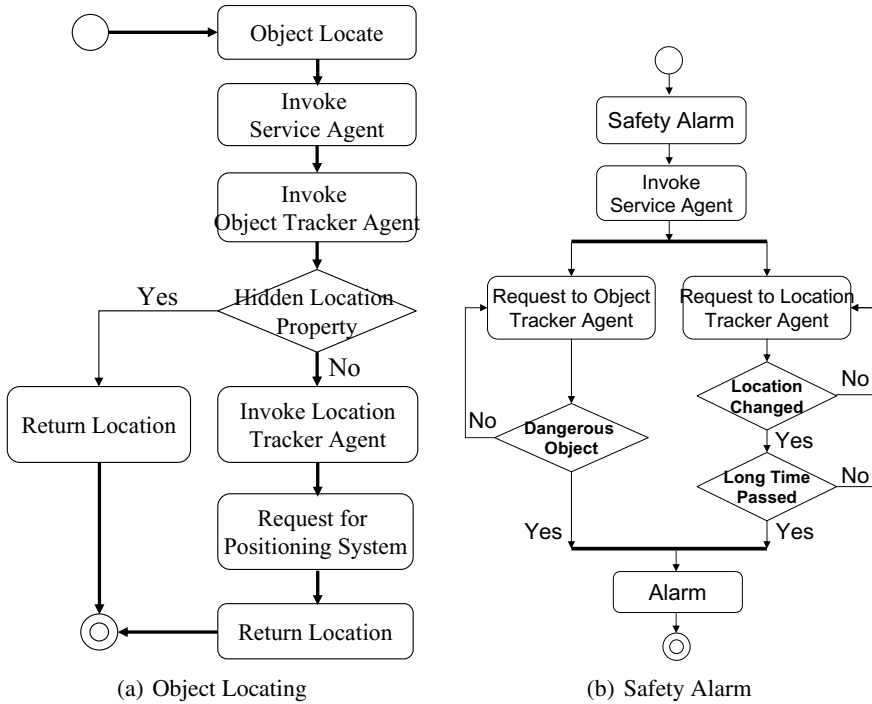


Fig. 5. Activity Diagram for Object Locating and Safety Alarm Services

**Agents in Safety Alarm Service.** People may ask the service of Safety Alarm to guard their safety in home or other places. The Service Agent will invoke Object Tracker Agent to monitor whether the object user touched has any potential danger, such as steam iron or gas burner. Figure 5(b) shows detail of agents’ activities for this service.

### 4 Experiments

We conducted our experiments in the 3rd floor of the Computer Science and Information Engineering at National Taiwan University. And we put everyday objects with RFID tag in our laboratory. Besides, we are equipped with wearable item with RFID reader, and use RS232 transmission line to connect our personal server. Figure 6 shows the equipments and objects we touched in the experiment.





**Fig. 6.** Equipments and objects we touched in the experiment.

#### 4.1 Object Recognition

We tested several objects when we touched the object in a certain condition. First, we attached lots of tags to our everyday objects. Objects include camera, microwave switch, book, wallet, PDA, mobile phone, mouse, pitcher, bowl, and spoon. These objects are placed in our laboratory. We used our wearable item with RFID reader and PDA as our sensing devices. We tested 20 times for each object.

Table 2 shows the experimental results for object identification. If the object is small enough to hold in hand, the precision will be higher, such as PDA, mobile phone and spoon. In order to have better performance, we suggest that the tag must be attached

**Table 2.** Precision for tagged object reads

Object Name	Test Condition	Tag Position	Success Reads	Precision (%)
Digital camera	Taking pictures	Left hand side	18	90
Microwave switch	Turning on	Switch	8	40
Book	Reading	Cover	16	80
Wallet	Retrieving money	Back	14	70
PDA	Holding	Back	18	90
Mobile phone	Talking	Back	19	90
Mouse	Web browsing	Middle	14	70
Pitcher	Holding	Side	11	55
Bowl	Drinking	Side	10	50
Spoon	Spooning up	Handle	19	95

to the objects according to human's behavior. The performance of object recognition is decided by the human's behavior and the interference of radio frequency.

## 4.2 The Problem in Location Tracking

We tested our Location Tracker Agent in the 3rd floor of the Computer Science Information Engineering building at National Taiwan University. We find several factors that will influence our results.

EPE 2.1 uses Wi-Fi networking technology, and the signal strength is not stable. Numbers of people, objects changed in the environment and the geography of access points will influence the signal strength that the client received. So the estimation of location position may have inaccuracy. If the rectangle of an area is too small, and the EPE 2.1 is unstable, then the location of device will change easier in different area. Therefore the size of our rectangle area is also an important issue in our research domain. Too big would lead user into a confusion about the location, and too small, the change would give user error information. Sometimes, the device doesn't change its location, but the result shows that it changed. Some areas like hallway (shown in Fig 2) have this problem seriously, because hallway is a narrow area, the system may misunderstand it as a room easily.

## 5 Related Work

People already use many kind of assistant tools for managing their daily routines and schedule. Reminder tools like to-do list or post-it note are very common used in our daily life to assist people to remember what they need to do next or bring something they forget. While tasks and plans are getting more and more complex, technology like personal information manager (PIM) or electronic calendar, can help us easily handle complicated and numerous personal objects. When we are now in pervasive computing environment, taking advantages of rich contexts provides such reminder system more features and intelligence. CybreMinder [1] is an early research of using context information for delivering messages via different ways in right situation. Messages can be voice message, e-mail or displaying on nearby displays. The user context here can be a person, location, time, activity, etc. In the similar way, Gate Reminder [2] also tries different reminding ways through different kind of interfaces deployed in a future smart home, as its name, this project sets up a home appliance located in the front door to achieve transparent interaction using RFID technology. The RFID reader will real time capture people and objects identification by passive tags, match home member's item list with their predefined schedule, then provide the effective reminders according to the right contexts. In their work, a series of design principle and user studies were given.

Another kind of application aims to use reminder as the assisted cognition tool for caring the elders and people with memory impairment. Autominder [3] is a project to support activity reminder and help clients remain their daily living activities, such as eating, performing hygiene, taking medicine and toileting. It keeps the client's activity model for further planning and has scalability for addition or modification of an activity. This system can be further implemented on a nurse robot for home care.

While wearable computer and sensor technology are easily deployed and embedded into system, collecting contexts is thus no more a complicated mission. One of the most potential topic of context inference is to infer human activity. A possible approach is to detect person-object interaction with a wearable device [4]. Recently, many researchers interest on exploiting RFID technology for person and object identification, furthermore, they can track what they have identified. The main concept is to perceive interaction between human and physical world [5]. Schmidt et al [6] suggest a wearable RFID-tag reader to handle tagged objects among explicit human-computer interaction. Pederson [7] has a similar device worn on his finger to track knowledge work actions in the office environment. The difference from his work and former systems is that he considers the location context, but sensing range is restricted in the local office.

The Guide project develops the iGlove [8, 9], a glove-based RFID reader with wireless communication, to infer activities of daily living. They successfully infer different types of activity from object-touch sequence. However, they do not combine location information to enhance the inference accuracy. The closest to our approach in the application level is the work developed by Borriello et al [10]. They try to remind people of misplace objects by monitoring them with two long range RFID readers housing in the door way. The personal server will receive item contexts from two readers and check through the personal data stored on it when user crossing through the door. In addition, the location module predicts the possible location destination from many inputs, e.g. calendar, user's schedule and objects sensed. Then, the reminder application generates reminders depending on rules.

## 6 Conclusion and Future Work

An intelligent alert system has been proposed to provide object reminder and safety alarm services. In our sample scenarios, potentially dangerous objects, such as gas burner and iron, are tagged. We have implemented the capability to track when and where any tagged object was touched. For example, a user may go to the kitchen to cook or go to the laundry room to iron his clothes. He may be interrupted by unrelated events like an "incoming telephone call" or a "door bell". The user may be distracted from what he was doing before and forget to turn off the gas burner or to unplug the iron in time. Our system is designed to reduce such safety risks.

We plan to add landmarks in our experimental environment by attaching RFID tags to object that are not easily movable, e.g. heavy furniture or fixtures. When the user touches any landmark, its pre-defined position will substitute for the coarse position obtained from Ekahau. For example, given a drinking fountain as a landmark in our lab r335, the Location Tracker Agent will obtain the precise location from the landmark when a user touches the drinking fountain. As a result, we know the user is in r335 for sure. Furthermore, we plan to infer additional activities by tracking interactions among users based on their locations detected by Ekahau.

## Acknowledgements

This work is partially supported by the National Science Council of Taiwan, R.O.C, under grant NSC-93-2218-E-002-148.

## References

1. Dey, A.K., Abowd, G.D.: Cybreminder: A context-aware system for supporting reminders. In: Proceedings of the 2nd international symposium on Handheld and Ubiquitous Computing (HUC '00), London, UK, Springer-Verlag (2000) 172–186
2. Kim, S.W., Kim, M.C., Park, S.H., Jin, Y.K., Choi, W.S.: Gate reminder: A design case of a smart reminder. In: Proceedings of the 2004 conference on Designing interactive systems (DIS '04), New York, NY, USA, ACM Press (2004) 81–90
3. Pollack, M.E., Brown, L., Colbry, D., McCarthy, C.E., Orosz, C., Peintner, B., Ramakrishnan, S., Tsamardinos, I.: Autominder: An intelligent cognitive orthotic system for people with memory impairment. *Robotics and Autonomous Systems* **44** (2003) 273–282
4. Fishkin, K., Jiang, B., Philipose, M., Roy, S.: I sense a disturbance in the force: Long-range detection of interactions with RFID-tagged objects. In: Proceedings of Sixth International Conference on Ubiquitous Computing (UbiComp 2004). (2004) 268–282
5. Want, R., Fishkin, K.P., Gujar, A., Harrison, B.L.: Bridging physical and virtual worlds with electronic tags. In: Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '99), New York, NY, USA, ACM Press (1999) 370–377
6. Schmidt, A., Gellersen, H.W., Merz, C.: Enabling implicit human computer interaction - a wearable rfid-tag reader. In: Proceedings of International Symposium on Wearable Computers (ISWC2000). (2000) 193–194
7. Pederson, T.: Magic touch: A simple object location tracking system enabling the development of physical-virtual artefacts in office environments. *Personal and Ubiquitous Computing* **5** (2001) 54–57
8. Philipose, M., Fishkin, K.P., Fox, D., Kautz, H., Patterson, D., Perkowitz, M.: Guide: Towards understanding daily life via auto-identification and statistical analysis. In: Proceedings of UbiHealth 2003: The 2nd International Workshop on Ubiquitous Computing for Pervasive Healthcare Applications. (2003)
9. Philipose, M., Fishkin, K.P., Perkowitz, M., Patterson, D.J., Fox, D., Kautz, H., Hhnel, D.: Inferring activities from interactions with objects. *IEEE Pervasive Computing* **3** (2004) 50–57
10. Borriello, G., Brunette, W., Hall, M., Hartung, C., Tangney, C.: Reminding about tagged objects using passive rfids. In: Proceedings of Ubiquitous Computing: 6th International Conference (UbiComp 2004), Springer (2004) 36–53

# Synchronization and Recovery in an Embedded Database System for Read-Only Transactions

Subhash Bhalla and Masaki Hasegawa

The University of Aizu, Aizu-Wakamatsu, Fukushima PO 965-8580, Japan  
{bhalla, d8041201}@u-aizu.ac.jp

**Abstract.** Transactions within an embedded database management system face many restrictions. These can not afford unlimited delays or participate in multiple retry attempts for execution. The proposed embedded concurrency control (ECC) techniques provide support on three counts, namely - to enhance concurrency, to overcome problems due to heterogeneity, and to allocate priority to transactions that originate from critical host.

## 1 Introduction

An embedded system is most often dedicated to a single application or small set of tasks. The software to manage them is small and simple. The operating conditions of the system are typically more restrictive than those of general purpose computing environments. An embedded system must continue to function without interruption and without administrator intervention.

In this paper, we focus on the Asilomar report "gizmo" databases [3]. These databases reside in devices such as smart cards, toasters, or telephones. The key characteristics of such databases are the following,

- the database functionality is completely transparent to users,
- explicit database operations or database maintenance is not performed,
- the database may crash at any time. It must recover instantly,
- the device may undergo a hard reset at any time. It requires that the database must return to its initial state, and
- the semantic integrity of the database must be maintained at all times.

As embedded systems define a specific environment and set of tasks, requiring expertise during the initial system configuration process is unacceptable. Many research efforts focus their attention on the maintenance of the system. For example, Microsoft's Auto Admin project [6], and the "no-knobs" administration. These have been identified as an area of important future research by the Asilomar authors [3].

## 2 Motivation - Embedded Database Systems

There are a few tasks that are typically performed by database administrators (DBAs) in a conventional database system. These tasks must be automated in an embedded system.

## 2.1 Outline of Requirements

Embedded systems typically perform simple queries. The relevant criteria are ease of maintenance, robustness, and small footprint. Of these three requirements, robustness and ease of maintenance are the more important criteria. Users must trust the data stored in their devices and must not need to manually perform anything resembling system administration in order to get their unit to work properly.

**Application Level Database System.** In an embedded database, the normal maintenance tasks must be automated. These are not necessarily based on the initial system configuration prepared by a user. There are five tasks that are traditionally performed by DBAs, but must be performed automatically in embedded database systems. These tasks are log archival and reclamation, backup, data compaction / reorganization, automatic and rapid recovery, and re-initialization from scratch.

Log archival and backup are tightly coupled. Database backups are part of any large database installation, and log archival is analogous to incremental backup [13]. There are a few implications of backup and archiving data in an embedded system. Consumers do not back up their VCRs or refrigerators, yet they back up their personal computers or personal digital assistants. We assume that backups, in some form, are required for gizmo databases (imagine having to reprogram, manually, the television viewing access pattern learned by some set-top television systems today) [13]. Furthermore, we require that those backups are nearly instantaneous or completely transparent, as users should not be aware that their gizmos are being backed up and should not have to explicitly initiate such backups.

Data compaction or reorganization has traditionally required periodic dumping and restoration of database tables and the recreation of indices. In an embedded system, such reorganization must happen automatically.

Recovery issues are similar in embedded and traditional environments with a few exceptions. While a few seconds or even a minute recovery is acceptable for a large server installation, no one is willing to wait for their telephone or television to reboot. As with archival, recovery must be nearly instantaneous in an embedded product. Secondly, it is often the case that a system will be completely reinitialized, rather than simply rebooted. In this case, the embedded database must be restored to its initial state, freeing all its resources. This is not typically a requirement of large server systems.

**System Level Architecture.** In addition to the maintenance-free operation required of the embedded systems, there are a number of requirements that fall out of the constrained resources found in the systems using gizmo databases. These requirements are: small footprint, short code-path, programmatic interface for tight application coupling and to avoid the overhead (in both time and size) of interfaces such as SQL and ODBC, application configurability and flexibility, support for complete memory-resident operation (e.g., these systems must run on gizmos without file systems), and support for multi-threading.

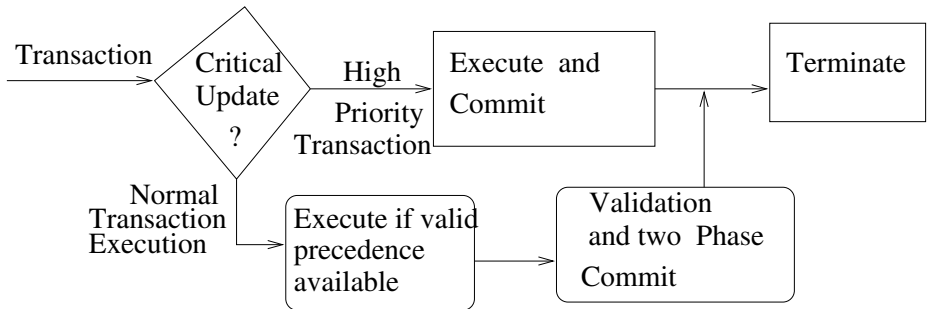
A small footprint and short code-path are common requirements for embedded database (EDB). However, traditional interfaces such as ODBC and SQL add significant size overhead and frequently add multiple context/thread switches per operation. These also add several IPC calls.

The rest of the manuscript is organized as follows. The next section describes database transactions. Section 4 presents a model of the system. It also describes the data sharing problems and a solution based on embedded concurrency control. Section 5 considers a proof of correctness. Section 6 presents related research activities. Finally, section 7 presents summary and conclusions.

### 3 Database Transactions

We consider, an environment based on transaction classification. The transactions at the server end are considered to be short and these can be easily restarted on account of few failures. The critical client's transactions on the other hand are considered instant execution requests of highest (real-time) priority. The server is assumed to have a high capacity and receives a few cases of critical client update requests. In many cases, the transaction processing system can execute a critical client (cc) update, with little or no overheads. In the study, conflicts among two critical client transactions are separately discussed at the end for sake of simplicity. We demonstrate the ease of processing a long read (backup) transaction using the proposed model.

In order to preserve serializability, the conventional systems depend on 2 phase locking (2PL) protocol [2]. Whereas the 2PL protocol enforces a two phase discipline, the criteria of serializability does not dictate the order in which a collection of conflicting transactions need to execute [2]. This option provides an opportunity to make a modified system that follows 2 PL protocol at the TM's level, but can be flexible at the data manager's (DM's) level. It can permit a interference free and 'non-blocked' execution for critical host (CH) transactions. This change necessitates maintaining 'lock table' in the form of site level graphs. Although this is the first effort (to the best of our knowledge) to use the technique for embedded databases, many graph based techniques have been studied earlier by [7], [11], [12].



**Fig. 1.** Execution of CH update transactions in isolation through embedded 2 phase locking based concurrency control

It is proposed to execute a critical host update (CHU) transaction in a special priority fashion. It may need to wait for another low-priority transaction, only if, that transaction has completed and local DM is participating in the second phase of a 2 phase commit.

The introduction of these possibilities integrates well with the existing transaction execution models. Earlier efforts at separating read-only transactions and update transactions exist [2]. The present study is an effort that proposes an implementation strategy for isolation of Serializable CHU transactions, for such an execution, that is free from interference by other transactions (Figure 1).

### 3.1 Transaction Execution

It is common for designers to extend the available approaches for concurrency control for use within the new system environments. However, we propose to study an analytical model and consider introduction of parallelism.

There have been some efforts at introducing parallelism within the concurrency control function. Earlier proposals attempt to eliminate interference between two classes of transactions. For example, processing Read-only transactions separately by using older versions of data, eliminate interference. Within the new classes, transactions are processed with no interference from each other's transactions. These can be considered to be executing in parallel. We propose to study the process of data allocation to executing transactions by using a stochastic process model. The model helps us in examining the parallel activity introduced by the use of classification of transactions. It also provides new insights that can lead to efficient processing of time-critical transactions. In the new environment, the time-critical transactions aim to execute with no interference from the ordinary transactions (Figure 1). In this light, the characteristics of the 2 Phase Locking based Concurrency Control scheme have been examined, within framework of a Real-Time (time-critical) database system.

## 4 The System Model

Based on the models of 2 phase locking and real-time computational environment with no slack time [9], a set of assumptions for executing transactions are organized. It is assumed that a 2 phase locking discipline is followed and the transaction execution is based on the criteria of serializability. Ideally, the CHU transactions should be able to do the following :

- a critical transaction may proceed without interference from other transactions.
- over ride conventional delays during execution
- integrate with existing modes of transaction executions. The two phases within the two phase locking ( 2PL ) protocol must execute with no blocking;
- execute and commit, i.e., if phase 1 is completed, then phase 2 needs to complete.

In the following section, a scheme to execute transactions as per a precedence order is described.



#### 4.1 Definitions : Embedded Database System

Embedded database system (EDS) consists of a set of data items ( say set 'D' ). The EDS is assumed to be based on a server that are occasionally accessed by critical hosts. The site supports a transaction manager (TM) and a data manager (DM). The TM supervises the execution of the transactions. The DMs manage individual databases. Each critical host supports a TM, that interacts with an EDB server. That performs other TM functions of interaction with other DMs. The network is assumed to detect failures, as and when these occur. When a site fails, it simply stops running and other sites detect this fact.

#### 4.2 The Transaction Model

We define a transaction as a set of atomic operations on data items. The system contains a mixture of instant priority real-time transactions (CHU, or CH reads) and ordinary transactions. We assume that the ordinary transactions can be aborted, in case of a data conflict with the real-time transactions.

The use of real-time database systems is growing in many application areas such as, industrial process control systems, and other time-critical applications. Many approaches for implementation of Real-Time systems are being studied [10]. In the real-time system environment, a critical transaction (computational task) is characterized by its computation time and a completion deadline. These systems are characterized by stringent deadlines, and high reliability requirements.

#### 4.3 Embedded Concurrency Control for Critical Data Operations

It is proposed to execute a CHT in isolation. It permits the CHT to proceed by locking data items. This step reduces the value of 'n' as the domain of locked items is confined to CHTs only. The norm for processing other transactions is based on an additional validation check, as per the criteria of serializability. For this purpose, each transaction is validated before commit.

The validation test for other transactions(OTs) uses the following criteria :

1. ( normal ) No data item, read by the transaction (OT), has been updated by a transaction after the read, that is -  
Read-set ( OT )  $\cap$  Write-set (more recently committed transactions ) ; and
2. ( additional ) No data item read by the transaction (OT), is in the locked item list of executing CHTs, that is -  
Read-set ( OT )  $\cap$  Locked-items ( CHTs ) .

The transactions that fail to meet the first criteria are aborted, and restarted. The transactions that fail to meet the second criteria can be made to delay commit, so as to let the executing CHT complete its execution. The algorithm for performing, the validation check is given below (Figure 2.). The possibility of repeated rollback of an ordinary transaction can be eliminated (It can be submitted as a low priority CHT). It can be observed that, such an execution introduces a non-interference environment for a CHT.

The above test is strictly conflict based and OTs need not perform any locking and can be made completely dependent upon a validation test [4].

```

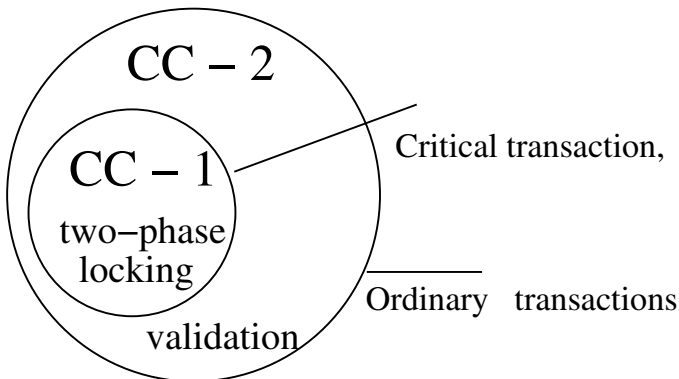
Procedure Validate (OT);
Valid := true;
For each X ∈ Read-set (OT) do
    if X ∈ write-set (Transactions committed after read by OT)
    then Valid := false, exit loop; end ;
    if X ∈ (Locked items list of CHTs )
    then wait for release and Valid := false, exit loop;
end;
If valid
then for each X ∈ write-set (T) do
    < Allot a commit sequence number to T >
    < Commit write-set (T) to database >
else restart (T);
end.
    
```

**Fig. 2.** Validation procedure for Embedded Concurrency Control ( ECC )

**Correctness Criteria.** The critical host update transactions execute as per the criteria of serializability by virtue of the 2PL protocol [2]. As the CHTs completely ignore the presence of OTs, these transactions are executed as per the notion of optimistic concurrency control, with an enhanced validation check. The validation check ensures that an OT is Serializable with respect to,

1. the previously committed transactions, and
2. the executing critical host transactions.

**Performance Considerations.** A drawback associated with adoption of validation based approaches is the possibility of repeated rollbacks. However the proposed scheme can prevent these rollbacks by resubmitting a rejected trans-



**Fig. 3.** Performance gain for Critical Transactions

action as a low priority CHT. This will make the OT execute as per the 2PL protocol and prevent the repeated rollbacks.

Although, in this approach the OTs face an enhanced level of validation, there are three positive aspects that are associated with our proposal. Firstly, the OTs avoid the problem of repeated rollbacks. Also, such a mixed mode of execution enhances the overall level of concurrency, because unlike locking based approaches the validation check is based on testing conflicts using the read-set of the committing transaction. Finally, the main item under focus concerning the is performance is the execution of CHT. Although the overall level of concurrency is expected to be improved, the performance of CHTs is enhanced as these execute with no interference with the other transactions within the system (Figure 3). A performance evaluation study has been presented in [5].

#### 4.4 Incremental Corrections to Global Read Contents

We propose an algorithm based on asynchronous computing ([4]). This algorithm has two stages. In the first stage, the log of the transactions is recorded during global-read. In the next stage, the backup copy is corrected by using the log of the transactions in off-line mode. The entity conditions and the normal transactions work the same as the original incremental global-read algorithm [1]. In this proposal, there are no rejected transactions.

In the proposed algorithm, a global read transaction is started. It locks a small part of the database as it proceeds. The database items that are read by the global read are colored as black. Other transactions that update data in the database are colored based on the items accessed by them. Transactions that read black data items are colored as black. Other transactions are colored as white or grey (mixed read-set). These transaction update database. The database items updated by the colored transactions are colored as grey. At the time of commit, if an entity's color is black and is updated by a black or gray transaction, then its contents are noted by using the log of the transactions. Later, the copied version of database (inconsistent version read by global read copy) is corrected by using the log (in off-line mode) (Figure 4.). This proposal creates a complete backup of the database which is consistent with the time, when the global-read is completed. This proposal does not need more storage, as a small log can be maintained in the main memory. Also, if a separate system recovery log is prepared for recovery by the system. By combining the backup log with system recovery log, no additional storage is needed.

**The Algorithm - Transformation of Database States.** As shown in Figure 4, phase 1, generates a modified log during the execution of a global-read transaction. This log (called the **color log**) contains a color marking for each update transaction. On completion of the global-read, the data read by the global-read contains an inconsistent version of the database. In phase 2, modifications are applied to make the data consistent, as shown in figure 4. An algorithm to generate the color log and for later generation of a consistent database version is described in this section. The various aspects of the proposed scheme are discussed below.

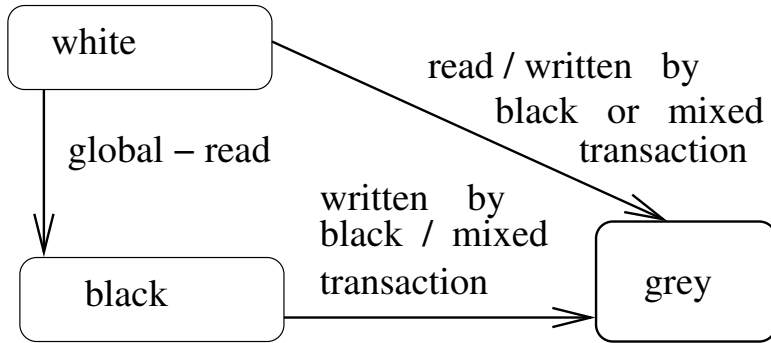


Fig. 4. State transition diagram for data entities

**Global-Read Processing.** The global-read transaction ( $T_{gr}$ ) divides the database entities into black and white colors. At the beginning, all database entities are white. Gradually, entities read by a  $T_{gr}$  are colored black. In addition, all entities that are written by black or mixed transactions are colored as gray. White data entity that are read by black or mixed transaction are also colored as gray. Thus all data entities are colored as white, or black or gray. Normal update transactions are colored white or black depending on the color of data entities being updated by them. A transaction is termed as mixed (gray color), based on the following conditions,

- a transaction that updates a mixture of black and white data entities;
- a transaction that reads a data entity that is colored as gray; and
- a transaction that writes on a data entity that is colored gray.

The color of a transaction can be determined at the time of its commit by examining the color of data items in its, read-set and write-set. Normal read-only transactions are not colored and proceed normally subject to the constraints of the two-phase locking protocol.

## 5 Proof of Correctness

While the earlier proposals avoid inconsistency by not allowing certain transactions to commit, our proposals permit a normal execution activity during the execution of the global-read transaction. All such updates that could have been missed partially or fully, are rewritten on the database copy, during phase 2. These updates by gray or black transactions, can generate inconsistency. Concurrent updates by white transactions are read by the global-read transaction, during global-read.

## 6 Related Work

Occasionally, leading researchers in the database community convene to identify future directions in database research. The most recent of discussion is the

1998 Asilomar report. It identifies the embedded database as one of the important research areas in database research [3]. Also, market analysts identify the embedded database market as a high-growth area in the commercial sector as well [8].

The Asilomar report identifies a new class of database applications, which they term "gizmo" databases, small databases embedded in tiny mobile appliances, e.g., smart-cards, telephones, personal digital assistants. Such databases must be self-managing, secure and reliable. Thus, the idea is that gizmo databases require plug and play data management with no database administrator (DBA), no human settable parameters, and the ability to adapt to changing conditions. More specifically, the Asilomar authors claim that the goal is self-tuning, including defining the physical DB design, the logical DB design, and automatic reports and utilities [3]. To date, few researchers have accepted this challenge, and there few research studies on the subject [13].

## 7 Summary and Conclusions

In a embedded database system, transactions need to backup data and perform updates. A possibility is demonstrated by considering the critical client, as a host issuing update (or read) transactions. This class of transactions can be executed as an instant priority real-time transaction with no slack time available. By adopting transaction classification, many changes can be accommodated within the conventional locking at a low cost that enable the database updates by critical clients.

## References

1. P. Amann, Sushil Jajodia, and Padmaja Mavuluri, "On-The-Fly Reading of Entire Databases", *IEEE Transactions of Knowledge and Data Engineering*, Vol. 7, No. 5, October 1995, pp. 834-838.
2. Bernstein P.A., V.Hadzilacos and N.Goodman, *Concurrency control and recovery in database systems*, Addison-Wesley, 1987.
3. Bernstein, P., Brodie, M., Ceri, S., DeWitt, D., Franklin, M., Garcia-Molina,H., Gray, J., Held, J., Hellerstein, J., Jagadish, H., Lesk, M., Maier, D., Naughton, J., Pirahesh, H., Stonebraker, M., Ullman, J., "The Asilomar Report on Database Research", *SIGMOD Record*, Vol. 27, No. 4,pp. 74-80, 1998.
4. Bhalla S., "Improving Parallelism in Asynchronous Reading of an Entire Database", *Proceedings of 7th High Performance Computing (HiPC 2000) conference*, Bangalore, December 2000, published by LNCS vol. 1970.
5. Bhalla S., "Asynchronous Transaction Processing for Updates With no Wait-for State", *Proceedings of 9th High Performance Computing (HiPC 2002) conference*, Bangalore, December 2002, published by LNCS vol. 2552.
6. Chaudhuri, S., Narasayya, V., "An Efficient, Cost-Driver Index Selection Tool for Microsoft SQL Server," *Proceedings of the 23rd VLDB Conference*, Athens, Greece, 1997.
7. Eich, M.H., and S.H.Garard, "The performance of flow graph locking", *IEEE Transactions on Software Engineering*, vol. 16, no. 4, pp. 477-483, April 1990.

8. Hostetler, M., "Cover Is Off A New Type of Database," *Embedded DB News*, <http://www.theadvisors.com/embeddeddbnews.htm>, 5/6/98.
9. Korth H.F., E. Levy, and A. Silberschatz, "Compensating Transactions: a New Recovery Paradigm," *Proceedings of 16th International Conference on Very Large Databases (VLDB)*, Brisbane, Australia, 1990, pp. 95-106.
10. Ramamritham K., "Real-Time Databases", *Distributed and Parallel Databases*, Kluwer Academic Publishers, Boston, USA, Vol. 1, No. 1, 1993.
11. Reddy P. Krishna, and S. Bhalla, "A Nonblocking Transaction Data Flow Graph Based Protocol For Replicated Databases", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 7, No. 5, October 1995.
12. Reddy P. Krishna, and S. Bhalla, "Asynchronous Operations in Distributed Concurrency Control", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 15, No. 3, May 2003.
13. Seltzer, M., and M. Olson, "Challenges in Embedded Database System Administration", May 2005, <http://www.sleepycat.com/docs/ref/refs/embedded.html>

# Learning with Data Streams – An NNTree Based Approach

Qiangfu Zhao

The University of Aizu, Aizuwakamatsu, Japan  
qf-zhao@u-aizu.ac.jp  
<http://www.u-aizu.ac.jp/qf-zhao>

**Abstract.** The plasticity-stability dilemma is a key problem for learning with data streams. On the one hand, the learner must be plastic enough to adapt to the new data. On the other hand, it must be stable enough to integrate information. In this paper, we try to resolve this problem using neural network trees (NNTrees). An NNTree is a decision tree (DT) with each non-terminal node containing an expert neural network (ENN). The NNTrees are plastic because they can adapt to the new data through retraining of the ENNs and/or through generation of new nodes. The NNTrees are also stable because retraining is performed partially and locally. In this paper, we propose an algorithm that can grow NNTrees effectively and efficiently. Experiments with several public databases show that the NNTrees obtained by the proposed methods are comparable with the NNTrees or DTs obtained with all data provided all-at-once.

**Keywords:** Machine learning, neural networks, decision trees, on-line learning, neural network trees, plasticity-stability dilemma.

## 1 Introduction

The plasticity-stability dilemma is a key problem for learning with data streams. On the one hand, the learner must be plastic enough to adapt to the new data. On the other hand, it must be stable enough to integrate information. Usually, fully connected neural networks are plastic but not stable. To make the system stable, some kind of localization is necessary. In this paper, we try to resolve this problem using neural network trees (NNTrees). An NNTree is a decision tree (DT) with each non-terminal node containing an expert neural network (ENN). The basic structure of an NNTree is shown in Fig. 1. The NNTrees are plastic because they can adapt to the new data through retraining of the ENNs and/or through generation of new nodes. The NNTrees are also stable because retraining of the ENNs is performed partially and locally. Partial retraining means that the weights of the ENNs are updated only for a limited number of epochs. Local retraining means that, for each new datum, only the ENNs on the “search path” are modified.

Note that traditional DTs can also be induced using a data stream through restructuring of the tree [1]. The problem in this approach is that all information

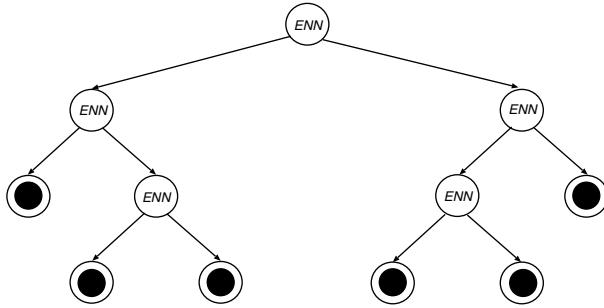


Fig. 1. Structure of an neural network tree

for restructuring the tree must be preserved. This is not possible if the duration of the data stream is very long. Using the NNTrees, however, we do not have to keep all information. As shown in this paper, a sliding window might be good enough to keep information for retraining the ENNs or for generating new nodes.

The main contribution of this paper is to propose an NNTree based algorithm for learning with data streams. The basic idea of this algorithm is to retrain the ENNs to adapt to new data. Retraining is performed partially in two senses. First, only a limited number of epochs are used in retraining. Second, only a smaller number of data kept in a sliding window are used in retraining. New nodes can be generated when retraining alone is not enough. This paper is organized as follows. In Section 2, we provide a brief review of some basic concepts related to DT. Section 3 proposes the new learning algorithm in detail. The efficiency and efficacy of the new algorithm is verified through experiments with several public databases in Section 4. Section 5 is the conclusion.

## 2 Review of Decision Trees

### 2.1 The Definition

A decision tree (DT) is a directed graph with no cycles. There is one special node called root. We usually draw a DT with the root at the top (see Fig. 1). Each node (except the root) has exactly one node above it, which is called its *parent*. The nodes directly below a node are called its *children*. A node is called a *terminal node* if it does not have any child. A node is *non-terminal* if it has at least one child. The node of a DT can be defined as a 5-tuple as follows:

$$node = \{I, F, Y, N, L\} \quad (1)$$

where  $I$  is a unique number assigned to each node,  $F$  is a test function that assigns a given input pattern to one of the children,  $Y$  is a set of pointers to the children,  $N = |Y|$  is the number of children or the size of  $Y$ , and  $L$  is the class label of the node if it is a terminal node. For terminal nodes,  $F$  is not defined and  $Y$  is empty ( $N=0$ ). Usually,  $L$  is not defined for non-terminal nodes.



The process for recognizing an unknown pattern  $x$  is as follows:

- Step 1: Set the root as the current node.
- Step 2: If the current node is a terminal node, assign  $x$  with the class label of this node, and return; otherwise, find  $i = F(x)$ .
- Step 3: Set the  $i$ -th child as the current node, and return to Step 2.

## 2.2 Induction of the DTs

To induce a DT, it is assumed that a training set is available. Usually, the DT is induced by partitioning the training set recursively. This procedure involves three steps: 1) splitting nodes, 2) determining which nodes are terminal nodes, and 3) assigning class labels to terminal nodes.

To see a node is a terminal node or not, the simplest way is to check if all or most examples assigned to this node belong to the same class. If all or most examples are from the same class, the node is terminal, and its label is usually defined as the class label of the majority examples.

The purpose of splitting a node is to find a good test function  $F$  for that node, so that the training examples assigned to this node can be partitioned into  $N$  groups according to the test results. Here we need a measure to quantify the “goodness” of the test function. Many criteria have been proposed in the literature [2]-[6]. For example, the criterion used in the well known induction algorithm C4.5 is the information gain ratio [6]. It is known that the performance of a DT dose not appear to vary significantly over a wide range of criteria [2].

## 2.3 Definition of the NNTrees

An NNTree is a DT with each non-terminal node containing an ENN (see Fig. 1). To use an NNTree for learning with data streams, we define a node of an NNTree as follows:

$$node = \{I, F, Y, N, L, W\} \quad (2)$$

where  $W$  is a sliding window to keep examples for retraining. The sliding window is nothing but a queue (or a first-in-first-out memory). The test function  $F$  is now defined by an ENN. As the ENNs, we use three layer multilayer perceptrons (MLPs). There are  $M$  inputs,  $K$  hidden neurons and  $N$  output neurons.

Using an NNTree, an example  $x$  can be recognized as follows:

- Step 1: Set the root as the current node.
- Step 2: If the current node is a terminal node, assign  $x$  with the class label of this node, and return; otherwise, find

$$F(x) = i = \arg \max_{1 \leq k \leq N} o_k \quad (3)$$

where  $o_k$  is the  $k$ -th output of the ENN.

- Step 3: Set the  $i$ -th child as the current node, and return to Step 2.

### 3 Learning with Data Streams Using the NNTrees

A direct method for learning with data streams is to fix the structure of the NNTree, and retrain the ENNs only. This method, however, cannot get good results because retraining the ENNs alone is not powerful enough to integrate new information effectively [10]. To improve the learnability of an NNTree, we proposed a new algorithm in this paper. Fig. 2 shows the flow-chart of this algorithm. The learning process is described as follows:

- Step 0: Initialize the tree by putting the first training example into the sliding window  $W$  of the root node, and define the class label of the root node  $s$  that of the first example. The tree now contains only one terminal node.
- Step 1: Set the current node as the root node and receive a training example  $x$  with the class label  $label(x)$  from the data stream.

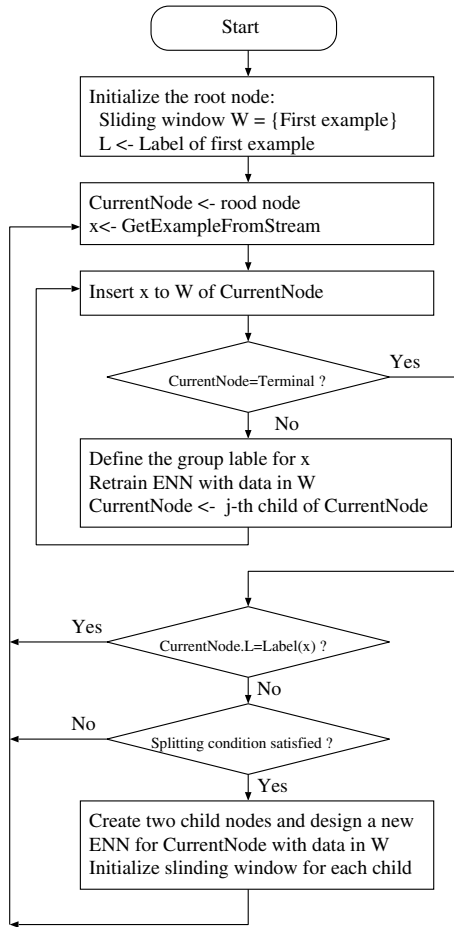


Fig. 2. Flow chart of the algorithm for learning with a data stream

- Step 2: Insert  $x$  into the sliding window  $W$  of the current node.
- Step 3: If the current node is terminal, goto Step 5; otherwise, continue.
- Step 4: Define the group label of  $x$  (see the definition given below), retrain the ENN using the back propagation (BP) algorithm for a limited number of epochs with data in  $W$ . After retraining, if the  $j$ -th output of the ENN is the maximum for  $x$ , set the current node as the  $j$ -th child of the current node, and return to Step 2.
- Step 5: If the  $L$  value of the current node is  $label(x)$ , return to Step 1; otherwise, continue (do nothing if  $x$  can be recognized correctly).
- Step 6: If the splitting condition (see the definition given below) is not satisfied, return to Step 1; otherwise, continue.
- Step 7: Create  $N$  child nodes and one ENN for the current node, and return to Step 1. The ENN is designed using BP algorithm with data in  $W$ . Based on the outputs of the ENN so designed, data in  $W$  are split into  $N$  parts. Data of the  $i$ -th part are put into the sliding window of the  $i$ -th child.

The group label of a example  $x$ , denoted by  $g(x)$ , is  $i$  ( $i = 1, 2, \dots, N$ ) if this example is assigned to the  $i$ -th child node of the current node. Note that  $g(x)$  is different from the class label of  $x$ . The class label  $label(x)$  is given before learning (for supervised learning), while  $g(x)$  must be determined during learning. In the proposed algorithm, the  $g(x)$  is found as follows

1. Find  $k = F(x)$ , where  $F(x)$  is the function realized by the ENN of the current node.
2. If  $\exists y \in W_k$ , where  $W_k$  is the sliding window of the  $k$ -th child, such that  $label(y) = label(x)$ , then  $g(x) = k$ .
3. Else, if  $\exists y \in W_j$  ( $j \neq k$ ), such that  $label(y) = label(x)$ , then  $g(x) = j$ .
4. Else,  $g(x) = k$ .

Once the group label of  $x$  is defined, we can retrain the ENN in the current node for a limited number of epochs. Note that the number of epochs is one when the ENN can assign  $x$  to the correct child (in the sense defined by  $g(x)$ ). Note also that if  $x$  is a completely new example, i.e., we have not observed any example of the class  $label(x)$  before, it will be assigned to the  $k$ -th child, where  $k = F(x)$ .

To prevent the tree from growing too fast, we use a splitting condition to determine when to split the node (in Step 6 of the learning algorithm). The condition used in our study is defined as follows:

$$|S| > T \text{ and } n_{wrong} > |S| \times s \quad (4)$$

where  $S$  is the set of examples assigned to the current node so far,  $|S|$  is the size of  $S$ ,  $n_{wrong}$  is the number of mis-classified examples,  $T$  is a threshold and  $s$  is the splitting-rate. Generally speaking,  $T$  and  $s$  depend on the training set size and the number of classes. In this study, we just set  $T$  to 30 and  $s$  to 0.1. With these values, the above condition can be read as “the current node will not be split if  $|S|$  is less than 30, or if the mis-classified examples are less than 10% of  $S$ ”. Note that  $S$  is not preserved during learning. Only  $|S|$  is used.

**Table 1.** Parameters of the Databases

	Number of examples	Number of features	Number of classes
cancer	683	9	2
crx	690	15	2
dermatology	358	34	6
ionosphere	351	34	2
iris	150	4	3
vehicle	846	18	4
optdigits	5620	64	10
pen-based	10992	16	10

When a terminal node is going to be split, we must design a new ENN for this node, and partition the data in the sliding window into  $N$  groups based on this ENN. In this paper, the data are grouped as follows. Suppose that we want to partition the data in  $W$  into  $N$  sub-sets  $S_1, S_2, \dots, S_N$ , which are initially empty sets. For any given example  $x$ ,

1. If there is a  $y \in S_i$ , such that  $label(y) = label(x)$ , assign  $x$  to  $S_i$ ;
2. Else, if there is a  $S_i$ , such that  $S_i = \Phi$ , assign  $x$  to  $S_i$ , where  $\Phi$  is the empty set;
3. Else, find  $y$ , which is the nearest neighbor of  $x$  in  $\cup S_i$ , and assign  $x$  to the same sub-set as  $y$ , where  $\cup$  represents the union of sets.

With the group labels so defined, we can design an ENN using the BP algorithm.

## 4 Experimental Results

To verify the efficiency of the proposed algorithm, we conducted several experiments using eight databases taken from the machine learning repository of the University of California at Irvine. The databases used include: cancer, crx, dermatology, ionosphere, iris, vehicle, optdigits and pen-based. These databases are actually real-life applications related to medical diagnosis, character recognition and so on. Parameters related to the databases are given in Table 1. 10-fold cross validation was used in the experiments. For learning with data streams, we select the training examples one-by-one without replacement, and provide it to the NNTree for retraining. The learning process will stop when all training examples are used.

The results of C4.5 [6] and ITI [1] are used for comparison. In the experiments, all parameters for C4.5 and ITI were default values. The main parameters related to the proposed method include: 1) the number of inputs  $M$  for each ENN is the number of features; 2) the number of hidden neurons  $K$  is 4; 3) the number of outputs  $N$  is 2/ 4) the maximum number of epochs for retraining the ENNs is 100; 5) the number of epochs for creating a new ENN is 1,000; and 6) the learning rate is 0.5.

The computer used in the experiments is a Unix workstation with the following specifications: 1) the system model is Sun Java Workstation W1100z; 2) the

main memory is 1 GB; 3) the CPU is 1.8GHz AMD Opteron 144 (1MB cache); and 4) the OS is Solaris 9. All algorithms were run on the same computer.

Note that C4.5 is an off-line learning algorithm. It assumes that all data are available before learning, and thus the results should be better in terms of accuracy when compared with the results obtained by ITI or the proposed algorithm. However, if we consider the computational cost for learning with data streams, C4.5 will be the worst if the duration of the data stream is long. This is because a new tree must be induced for each new example. Compare ITI with the algorithm proposed here, ITI must preserve all information for restructuring the tree during learning. Thus, for long data streams, the computational cost of ITI will be higher than that of the proposed algorithm.

Tables 2-5 are the results obtained by C4.5, ITI and the proposed method with different sliding window sizes. In the tables, “Tree size” is the number of all nodes, “Error rate (%)” is the rate for the test set, and “Computing time” is the time in seconds used for one run. For C4.5, only the results after pruning are provided here. For each result, there are two values. One is the average value over 10 runs, and another is the 95% confident interval.

#### 4.1 Discussion About the Accuracy

As expected, the accuracy of the DTs (decision trees) induced by C4.5 are the best for 4 cases out of 8. This is because that all data are available before learning. Although the authors of [1] claimed that ITI can obtain the same results as C4.5, the results given here do not support their claim. People may expect that NNTrees should outperform the DTs because they are more complex. This may be true for off-line learning. In this paper, however, we are talking about learning with data streams. That is, data are available only one by one. Results given here actually show that the NNTree based approach can resolve the plasticity-stability dilemma to some extent because the NNTrees are comparable with the DTs induced by C4.5 or ITI. In 3 of the cases they are the best.

#### 4.2 Discussion About the Size

In all cases, the number of nodes required by the NNTrees are smaller or much smaller than that of the DTs obtained by C4.5 or ITI. This is because an ENN is much more powerful than an axis-parallel hyperplane. Usually, one ENN can replace many hyperplanes for making the same decision. However, the tree size alone cannot be used to measure the complexity of the system. If we count the number of free parameters, the NNTrees are actually more complex. This increase in complexity is the main reason why NNTrees are able to learning with data streams.

#### 4.3 Discussion About the Computational Cost

The main computational costs of the proposed algorithm include: 1) cost for computing the output of the network for each example, 2) cost for updating the

**Table 2.** Results of C4.5

Database	Tree size	Error rate (%)	Computing Time
cancer	23.2±10.8	4.8±4.3	0.021±0.002
crx	30.3±24.3	14.6±7.2	0.035±0.003
dermatology	32.6±6.9	6.4±6.8	0.020±0.003
ionosphere	27.2±6.5	<b>10.8±8.7</b>	0.090±0.011
iris	8.2±2.0	<b>4.0±6.8</b>	0.016±0.002
vehicle	136.0±45.5	<b>28.5±8.5</b>	0.078±0.010
optdigits	410.0±29.4	9.4±2.1	0.966±0.061
pen-based	380.4±23.8	<b>3.5±1.0</b>	0.687±0.011

**Table 3.** Results of ITI

Database	Tree size	Error rate (%)	Computing Time
cancer	18.4±2.2	4.6±2.5	0.115±0.017
crx	28.6±4.4	<b>13.5±3.46</b>	0.618±0.039
dermatology	37.0±3.0	6.7±3.75	0.582±0.092
ionosphere	19.2±2.6	12.8±6.17	1.194±0.096
iris	5.8±0.6	3.8±4.4	0.012±0.004
vehicle	118.8±12.2	28.59±5.14	4.933±0.445
optdigits	919.2±53.8	22.3±1.6	510.639±65.885
pen-based	648.2±28.6	5.8±0.9	480.042±11.180

**Table 4.** Results of proposed method, window size=50

Database	Tree size	Error rate (%)	Computing Time
cancer	3.20±0.63	<b>4.1±1.8</b>	0.383±0.273
crx	17.40±3.86	17.4±4.1	5.604±2.190
dermatology	11.80±1.03	3.6±3.2	0.321±0.108
ionosphere	9.00±2.31	13.6±4.9	1.789±1.040
iris	6.00±1.05	5.0±5.7	0.352±0.237
vehicle	59.40±9.18	31.3±4.6	16.699±3.113
optdigits	41.00±5.25	7.8±1.9	24.052±5.316
pen-based	68.00±11.78	7.1±1.9	29.361±3.532

**Table 5.** Results of proposed method, window size=100

Database	Tree size	Error rate (%)	Computing Time
cancer	3.20±0.63	4.5±3.1	2.385±1.485
crx	14.20±2.53	16.7±3.6	14.614±2.474
dermatology	11.60±0.97	<b>2.2±2.9</b>	0.564±0.257
ionosphere	8.40±2.32	11.9±6.7	6.415±2.763
iris	5.60±0.97	4.4±5.2	0.803±0.477
vehicle	59.20±9.11	30.0±3.5	41.468±8.224
optdigits	39.80±7.07	<b>6.9±1.6</b>	82.543±22.161
pen-based	54.00±9.30	5.9±1.3	88.233±15.901

weights, 3) cost for all epochs, 4) cost for all nodes on the *search path*, and 5) cost for all training examples. Put all these costs together, we have the total cost:

$$Cost(Proposed) = O(M \times K \times N_w \times N_e \times \log T \times N_t)$$

where  $M$  is the dimensionality of the feature space,  $K$  is the number of hidden neurons,  $N_w$  is the number of examples in the sliding window,  $T$  is the size of the current tree, and  $N_t$  is the length of the data stream.

On the other hand, the computational cost of C4.5 is

$$\begin{aligned} Cost(C4.5) &= O[M \times m \times \log T \times (1 + 2 + \dots + N_t)] \\ &= O(M \times m \times \log T \times N_t^2) \end{aligned} \quad (5)$$

where  $m$  is the possible number of values of a feature and  $T$  is the size of the current tree. Note that the computational costs in all levels (not node) should be almost the same. Therefore, instead of using  $(T - 1)/2$ , which is the number of non-terminal nodes, we use  $\log T$  here. Note also that in the worst case,  $m = N_t$ , and thus

$$Cost(C4.5) = O(M \times \log T \times N_t^3)$$

Based on the above considerations, we can get the ratio between the cost of the proposed method and that of C4.5:

$$\begin{aligned} R &= \frac{O(M \times K \times N_w \times N_e \times \log T \times N_t)}{O(M \times \log T \times N_t^3)} \\ &= \frac{O(K \times N_w \times N_e)}{O(N_t^2)} \end{aligned} \quad (6)$$

In the experiments,  $N_e = 100$ ,  $K = 4$ , and  $N_w = 50$ , or 100. Thus, for long data streams, C4.5 will be much more time-consuming than the proposed method. Note that in Table 2, the computing time is only for the case  $n_t = N_t$ . If the data are provided one by one, the computing time should times the factor  $N_t(N_t + 1)/2$ .

As for the algorithm ITI, we do not know exactly the computational cost. In fact, even the authors of [1] did not provide an analysis of the computational cost. However, from the results given in Tables 3-5 we can see that for long data streams, ITI is more time consuming than the proposed method. This is also what we have expected.

## 5 Conclusion

In this paper, we have studied learning with data streams using the NNTrees, and proposed a new algorithm. The efficiency and efficacy of the algorithm have been verified through experiments with eight public databases. Note that all databases used in the experiments are not “endless data streams”. We will do

more experiments in the future using longer data streams to verify the usefulness of the proposed algorithm. In fact, we are trying to apply the proposed algorithm to stock prediction. More results will be reported later.

## Acknowledgments

This research is supported in part by the Grants-in-Aid for Scientific Research of Japan Society for the Promotion of Science (JSPS), No. 17500148.

## References

1. P. E. Utgoff, N. C. Berkman and J. A. Clouse, "Decision tree induction based on efficient tree restructuring," *Machine Learning*, vol. 29, pp. 5-44, 1997.
2. L. Brieman, J. H. Friedman, R. A. Olshen and C. J. Stong, *Classification and regression trees*, Belmont, CA: Wadsworth, 1984.
3. S. B. Gelfand, C. S. Ravishankar, and E. J. Delp, "An iterative growing and Pruning algorithm for classification tree design," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **13-2** (1991) 163-174.
4. E. G. Henrichon, JR. and K. S. Fu, "A non-parametric partitioning procedure for pattern classification," *IEEE Trans. on Computers*, **18-7** (1969) 614-624.
5. L. Hyafil, "Construction optimal binary decision trees is NP-complete," *Information Processing Letters*, **5-1** (1976) 15-17.
6. J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, 1993.
7. Q. F. Zhao, "Evolutionary design of neural network tree - integration of decision tree, neural network and GA," *Proc. IEEE Congress on Evolutionary Computation*, (Korea, Seoul, 2001). 240-244.
8. Q. F. Zhao, "Training and retraining of neural network trees," *Proc. INNS-IEEE International Joint Conference on Neural Networks*, (Washington DC, 2001) 726-731.
9. Q. F. Zhao, "Modeling and evolutionary learning of modular neural networks," *Proc. The 6-th International Symposium on Artificial Life and Robotics*, (2001) 508-511.
10. T. Takeda and Q. F. Zhao, "A two step algorithm for designing small neural network trees," *Proc. IEEE International Conference on Neural Networks and Signal Processing (ICNNSP03)*, (China, Nanjing, 2003) 513-517.
11. Q. F. Zhao, "Design Smart NNTrees Based on the  $R^4$ -rule," *Proc. of the IEEE 19th International Conference on Advanced Information Networking and Applications (AINA2005)*, Vol. 2, pp. 547-551, 2005.



# Generating Smart Robot Controllers Through Co-evolution

Kouichi Sakamoto and Qiangfu Zhao

The University of Aizu Aizuwakamatsu, 965-8580, Japan  
{d8061104, qf-zhao}@u-aizu.ac.jp

**Abstract.** To evolve robot controllers that generalize well, we should evaluate the controllers using as many environment patterns (evaluation patterns) as possible. However, to evolve the controllers faster, we should use as few evaluation patterns as possible in evaluation. It is difficult to know in advance what patterns can produce good controllers. To solve this problem, this paper studies co-evolution of the robot controllers and the evaluation patterns. To improve the effectiveness of co-evolution, we introduce fitness sharing in the population of evaluation patterns, and the inter-generation fitness in selecting good controllers. Simulation results show that the proposed method can get much better robot controllers than standard co-evolutionary algorithm.

## 1 Introduction

Neural network is a good model for robot control because it can acquire the control rules automatically through learning. In many cases, the rules cannot be given beforehand, because the environment may change frequently. In such cases, the robot must be smart enough to acquire the rules by itself. Since the teacher signals are often not available, supervised learning cannot be used. It is known that evolutionary learning or reinforcement learning is more efficient. In reinforcement learning, when the robot takes a certain action in the current state, there is some feedback (reward or penalty) from the environment, and the robot can learn based on the feedback. In evolutionary learning, only the final result is evaluated. In general, evolutionary learning uses much less information in learning. In this research, we consider evolutionary learning only.

There is one dilemma in evolving smart robot controllers. On the one hand, the evolution may become unstable if the environment of evolution is not fixed, and good controllers cannot be obtained. On the other hand, if the environment for evolution is fixed, the controller might be good only for that environment. That is, the robot is just a lucky-guy that cannot generalize well. One method for resolving this dilemma is to choose some typical evaluation patterns (environment patterns). The individual robot controllers can be evaluated using these patterns during evolution. Using this method, we can reduce the chance of obtaining lucky-guys because the robot controller should be good for different evaluation patterns. Also, because the evaluation patterns are fixed, the evolution can be stable. The problem is that in general it is difficult to choose the evaluation patterns that can generate good controllers.

In this paper, we try to resolve the above dilemma through co-evolution between the evaluation patterns and the robot controllers. We start from a simple problem first here. The problem considered is to evolve a mobile robot that can reach a given goal from any start point in an environment containing obstacles. The layout of the environment is not changed during evolution. The evaluation patterns in this case are the start points and the orientation of the robot. If we can solve this problem well, we can get some important hints for solving more complex problems in the next step.

When we use co-evolution, it is necessary to consider the following points:

1. Because we would like to obtain different evaluation patterns, fitness sharing is necessary for the population of the evaluation patterns. If we do not use fitness sharing, all evaluation patterns in the population will be very similar after evolution, and the robot so evolved will have a good chance to be the lucky-guy.
2. Since the robots and the evaluation patterns evolve together, when the evaluation patterns change, the criteria for evaluating the robots also change. If the robot individuals are selected based only on the fitness in the current generation, many good individuals can be selected against. This is a problem seen in dinosaur extinction. To solve this problem, we introduce the concept of inter-generation fitness, which is the sum or average fitness of an individual over several generations. The robot individuals are selected based on their inter-generation fitness.

In this paper, we will confirm the effectiveness of the above ideas through simulation experiments. This paper is organized as follows. In the next section, some preliminaries related to this work are provided. In Section 3, we give a short review on the GA (genetic algorithm) based evolution of the robot controllers. In Section 4, we describe the co-evolution in detail. Section 5 provides the simulation results, and Section 6 is the conclusion.

## 2 Preliminaries

### 2.1 The Robot Used in This Study

In this study, we use the Khepera robot which is a well-known mini mobile robot used by many researchers for studying intelligent robots. However, we do not use the real robot in our experiments because that will be very time consuming. We use the free software Khepera simulator 2.0. The simulation environment is a squared map of  $1000 \times 1000$  pixels. The layout of the robot(s), the obstacles, and the goal(s) can be defined by the user.

### 2.2 The Controller Model

The multilayer perceptron (MLP) is used as the robot controller in this study. The network has one input layer, one hidden layer and one output layer. The

input layer has 17 inputs (8 infrared sensors, and 8 light sensors and one bias). For convenience, the number of hidden neurons is also fixed to 17 without fine-tuning. Two output neurons are used to encode four actions:

1. Move forward: both outputs are larger than 0.5.
2. Move back: both outputs are less than 0.5.
3. Turn right: the first output is larger than 0.5, and the second output is less than 0.5.
4. Turn left: the first output is less than 0.5, and the second output is larger than 0.5.

### 3 Evolving the Robot Controllers Based on Standard GA

The genetic algorithm (GA) can be used for evolving the robot controllers. For this purpose, we need to define the fitness and genotype of the individuals. The genotype of a neural network robot controller is simply the list of all connection weights represented in real numbers. Since the problem considered here is to obtain robot controllers that can drive the robot to a given goal (light source) from any start point, we can define the fitness based on the distance between the robot and the goal after the robot moves for a certain number of steps. If the robot can reach the goal, we should prefer those that can reach the goal quickly. Specifically, the fitness of a robot controller can be calculated as follows:

1. The robot is put to the start points.
2. Let the robot move in the environment based on the sensor inputs and the decisions made by the controller. The robot stops when the number of moves reaches to 2,000, or when it reaches the goal.
3. If the robot cannot reach the goal, the fitness is defined as follows:

$$fitness = A \times (B - distance) \quad (1)$$

where A and B are constants, and distance is the distance between the robot and the goal when the robot stops. In this paper, we defined A=0.1 and B=700. If the robot reaches the goal within 2,000 steps, the bonus given below is added to the fitness.

$$Bonus = [(2000 - n)^2 \times C + D] \quad (2)$$

Here, C=0.000016 and D=14. The parameters A, B, C and D are so chosen that when the robot reaches the goal with 1,000 steps, the fitness is 100. If the robot reaches the goal with 2,000 steps, the fitness becomes 84. The fitness is still higher than the fitness when the robot cannot reach the goal.

## 4 Co-evolution of Robot Controllers and Evaluation Patterns

### 4.1 General Considerations

To evolve the robot controller using GA, the selection of start points is very important. If we fix one start point, the robot can reach the goal from this start points, but may not be able to reach goal if we put it to other start point. If we evaluate the robot using start points randomly generated in each generation, the generalization ability may become higher, but the evolution process may not be stable because the evaluation criterion is not constant. To solve this problem, we study the co-evolutionary approach here. The subjects to be co-evolved are the robot controllers and the start points (the evaluation patterns). The Individual of an evaluation pattern includes the  $x$  and  $y$  coordinates of the start point, and the orientation of robot at that point.

### 4.2 Standard Co-evolution

First, let us consider the standard co-evolution. Fig. 1 shows the flowchart of co-evolution. It can be described as follows:

- Step 0: Generate two populations, one for the robot controllers, and another for the evaluation patterns. All individuals are initialized using random numbers.
- Step 1: Evaluate the evaluation patterns using all robots. Specifically, for a given evaluation pattern, put all robots, one by one, to the position and orientation defined by this pattern, and let the robot move towards the goal. For each robot, we get a fitness value as defined by (1) and (2). The fitness of the evaluation pattern is defined as the sum of the fitness values of all robots. Note that fitness of an evaluation pattern should be as small as possible. That is, we should find such patterns from which the robots cannot reach the goal easily.
- Step 2: Evolve the evaluation patterns using standard GA defined in the previous section.
- Step 3: Evaluate the robots using all the evaluation patterns. For each robot, the fitness of the robot for one evaluation pattern is defined by (1) and (2). Its total fitness is the sum of the fitness values for all evaluation patterns. The fitness of the robots should be as high as possible.
- Step 4: Evolve the robots using standard GA.
- Step 5: If the terminating condition is satisfied, stop; otherwise, return to Step 1. In our experiment, the terminating condition is very simple. We just restrict the number of generations to 100.

### 4.3 Modified Co-evolution

In the standard co-evolution, each individual in one population must be evaluated by all individuals in another population. This is very time consuming. To

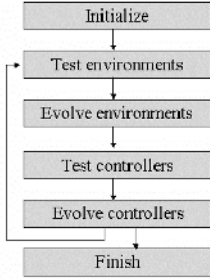


Fig. 1. Flowchart of standard co-evolution

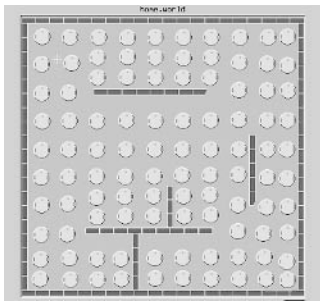


Fig. 2. Start points for testing the robot

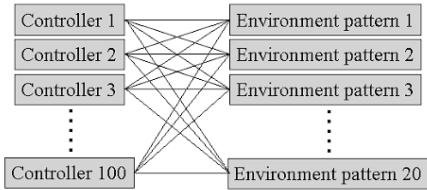


Fig. 3. Evaluation of the individuals in standard co-evolution

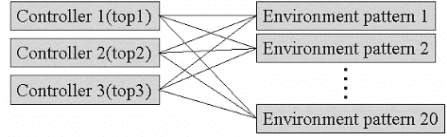


Fig. 4. Evaluation of the environment patterns in the modified co-evolutionary algorithm

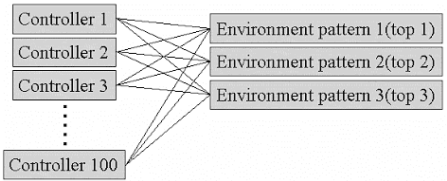


Fig. 5. Evaluation of the robots in the modified co-evolutionary algorithm

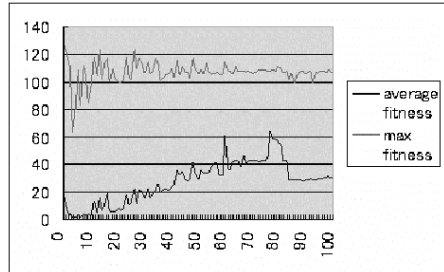


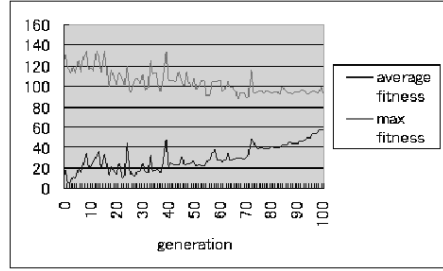
Fig. 6. Evolution curve of co-evolution with fitness sharing

speed-up the evolution process, we consider a modified co-evolution here. In the modified version, each individual is evaluated using several top individuals in another population. In this study, we just use the top 3 individuals in a population to evaluate the individuals in another population (Fig. 4 and Fig. 5). Note that in evaluating the robots, each environment pattern is used twice to make the evaluation results more reliable. Since there are some noises in the sensor inputs, the result for each evaluation is different.

In this study, the size of the robot population is 100, and that of the environment pattern population is 20. Therefore, the total number of evaluations in one generation in the standard co-evolution is  $100 \times 20 + 100 \times 20 \times 2 = 6,000$ . In

**Table 1.** Result of GA

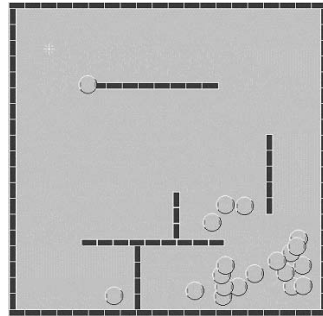
	Method 1	Method 2
Controller1	33	78
Controller2	52	53
Controller3	43	44
Controller4	48	71
Controller5	26	43
Controller6	32	79
Controller7	32	71
Controller8	26	73
Controller9	19	63
Controller10	52	74
Average	36.3	64.9



**Fig. 7.** Evolution curve of co-evolution with fitness sharing and inter-generation fitness

**Table 2.** Result of standard Co-evolution

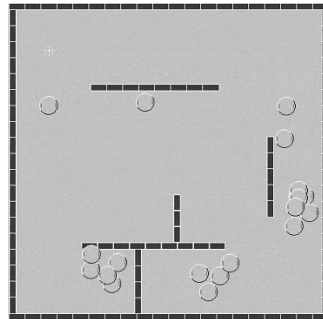
	Method 3	Method 4	Method 5
Controller1	61	75	56
Controller2	59	60	61
Controller3	55	57	56
Controller4	52	57	61
Controller5	56	84	87
Controller6	53	77	86
Controller7	52	75	84
Controller8	64	70	78
Controller9	61	59	59
Controller10	51	59	57
Average	56.4	67.3	70.1



**Fig. 8.** Evaluation patterns obtained by the standard co-evolution after 100 generations

**Table 3.** Result of modified Co-evolution

	Method 6	Method 7	Method 8
Controller1	49	64	76
Controller2	51	63	77
Controller3	74	72	74
Controller4	67	67	85
Controller5	59	71	74
Controller6	63	69	72
Controller7	68	65	74
Controller8	63	65	78
Controller9	69	86	73
Controller10	84	82	77
Average	64.7	70.4	76.0



**Fig. 9.** Evaluation patterns obtained by the co-evolution with fitness sharing after 100 generations

the modified co-evolution, the number of evaluations is  $100 \times 2 \times 3 + 20 \times 3 = 660$ . The speed-up ratio is about 9.

### 4.4 Fitness Sharing

There is an important problem in the above co-evolutionary algorithm. In fact, the evaluation patterns after evolution tend to be close to one point in the

environment. Usually, this point is the most difficult start point for the robot to reach the goal. Note that our purpose is to evolve robots that can reach the goal from ANY start point. Using only one point (although it is the most difficult one) cannot result in robot that generalizes well.

To solve the above problem, we introduce fitness sharing in the population of the environment patterns. For each individual, we first find its neighbors in a certain range. If there are  $N$  neighbors, its fitness is divided by  $N$ . In this study, we just adopt a hard limit for defining the neighborhood. For two points  $p_1$  and  $p_2$ , if their distance is less than  $R$ , we say that they are neighbors. In our simulations,  $R = 250$ . With fitness sharing, different environment patterns can be obtained through evolution.

#### 4.5 Inter-generation Fitness

Since our purpose is to find one robot that can go from any start point to the goal, we do not use fitness sharing in the robot population. However, there is another problem in evolving the robots. Because the evaluation patterns evolve together with the robots, good robots may suddenly become not so good or even bad in the next generation. This results in an unstable evolution. To solve this problem, we propose to use the inter-generation fitness in selecting the robot individuals. By inter-generation fitness we mean the total or average fitness of the robot over many generations. By so doing, robots that are good for different evaluation patterns generated in different generations can be preserved.

## 5 Simulation Results

To verify the ideas given in the previous section, we conducted several experiments. Results of the following methods are used for comparison:

- Method 1: Fix the start point to the center of the environment, and evolve the robot using GA;
- Method 2: Generate a new start point at random in each new generation, and evolve the robot using GA;
- Method 3: Evolve the robot and the evaluation pattern (the start point) together via standard co-evolution;
- Method 4: Standard co-evolution with fitness sharing;
- Method 5: Standard co-evolution with fitness sharing and inter-generation fitness based evaluation;
- Method 6: Modified (simplified) co-evolution;
- Method 7: Modified co-evolution with fitness sharing;
- Method 8: Modified co-evolution with fitness sharing and inter-generation fitness based evaluation.

To test the performance (generalization ability) of the robot controllers, we count the number of times the robot can reach the given goal from 100 start

points as shown in Fig. 2. The number of successes divided by the total number of test cases is called the success rate. To make the results more reliable, we evolved 10 robot controllers, and evaluated each controller 10 times.

The success rates are used to compare the effectiveness of the methods. The parameters used in the simulations are 1) The number of generations is 100 or 1,000; 2) The size of the population for robots is 100; 3) The size of the population for the environment patterns is 20; 4) The number of individuals (of another population) used for evaluation is all or 3; 5) Weight-by-weight mutation is used, with the mutation rate=0.002; 6) Two-point crossover with the crossover rate=0.8; and 7) Truncation selection with the selection rate=0.2.

Tables 1-3 are the experimental results. The numbers shown in the tables are the success rates (in %) of the robots. From these tables we can see that co-evolution without fitness sharing cannot generate good robot controllers. Method 2 is actually better. Fitness sharing can improve the generalization ability of both standard co-evolution and the modified one. However, from Fig. 6 we can see that fitness sharing alone is not enough to stabilize the evolutionary process. The (average) fitness of the population can drop sharply in some generation. Using inter-generation fitness, the evolution can be more stable (see Fig. 7).

Note that from Fig. 7 we can also see that the fitness of the best robot controller did not increase through evolution. This is because that the evaluation patterns are becoming more and more difficult. This is actually a relative measure. In this sense, the success rate shown in Tables 1-2 is the absolute measure.

One interesting fact is that the modified co-evolutionary algorithm is better than the standard one, although its computation cost is lower. We need to do more experiments to see if this is generally true or not.

Fig. 8 and Fig. 9 show the evaluation patterns obtained by the standard co-evolution and the co-evolution with fitness sharing. Clearly, co-evolution with fitness sharing can get different evaluation patterns, while standard co-evolution tends to provide evaluation patterns in the same location.

## 6 Conclusion

In this paper, we have investigated different co-evolutionary algorithms for evolving robot controllers. We found that fitness sharing and inter-generation fitness are very useful for improving the performance of the evolved robot controllers. However, the success rates are still not high enough. Further improvement is required to get better controllers.

In the future, we would like to use parallel computation to speed-up the evolution process first. Using parallel computation, we can increase the number of generations as well as the population sizes, and hopefully we can get better robot controllers. We will also try to solve more difficult problems such as evolving robots that can approach to any goal from any start point, and for any given environment layout. In addition, we would like to transform the learned results into understandable and re-usable rules.



## References

1. T. Endo and Q. F. Zhao, "Generation of comprehensible decision trees through evolution of training data," Proc. IEEE Congress on Evolutionary Computation (CEC'2002), pp. 1221-1225, 2002.
2. M. Ikarashi "Acquisition of Robot's Moving Strategies through Co-evolution" Master Thesis, The University of Aizu, March 2003.
3. L. M. Fu, "Rule generation from neural networks," IEEE Trans. System, Man, and Cybernetics, Vol. 24, No. 8, pp. 114-124, 1994.
4. Genetic algorithm with co-evolution mechanisms <http://www.symbiolab.sys.i.kyoto-u.ac.jp/research/coevga/coevga.htm>
5. K. Hirasawa, K. Nakanishi, T. Eguchi, & J. Hu "Multi Agent Systems with Symbiotic Learning and Evolution -Masbiole- and Its Application" IEEJ Transactions on Electronics, Information and Systems, Vol. 123-C, No.1, pp. 67-74, 2003.
6. T. Eguchi, K. Hirasawa, J. Hu, & J. Murata "Multi Agent Systems with Symbiotic Learning and Evolution using GNP" IEEJ Transactions on Electronics, Information and Systems, Vol. 123-C, No.3, pp. 517-526, 2003.
7. "Evolving controllers for a homogeneous system of physical robots:structured co-operation with minimal sensors" Matt Quinn, Lincoln Smith, Giles Mayley, Phil Husband
8. K. Sakamoto, T. Takeda and Q. F. Zhao, "Generation of good training data for extracting DTs from evolved NN robot controllers," Proc. IEEE International Conference on Neural Networks and Signal Processing (ICNNSP03), pp. 33-36, Dec. 2003.

# Integrated Multimedia Understanding for Ubiquitous Intelligence Based on Mental Image Directed Semantic Theory

Masao Yokota and Genci Capi

Department of System Management, Faculty of Information Engineering,  
Fukuoka Institute of Technology,  
3-30-1, Wajiro-higashi, Higashi-ku, Fukuoka-shi 811-0295 Japan  
{yokota, capi}@fit.ac.jp  
<http://www.fit.ac.jp>

**Abstract.** An ideal ubiquitous computing environment can be a network system of such intelligent and human-friendly robots that never appear in front of humans except when needed. In this paper the distributed intelligent robot network (DIRN) is proposed as one kind of wireless sensor and actor networks (WSAN) consisting of one brain node and numerous sensor and actor nodes with human-friendly interfaces. In order to realize well-coordinated DIRNs, it is very important to develop a systematically computable knowledge representation language universal for any kind of device as well as efficient networking technologies. As a candidate for this purpose, the multimedia description language  $L_{md}$  was evaluated by applying it to simulation of DIRN-world interaction.

## 1 Introduction

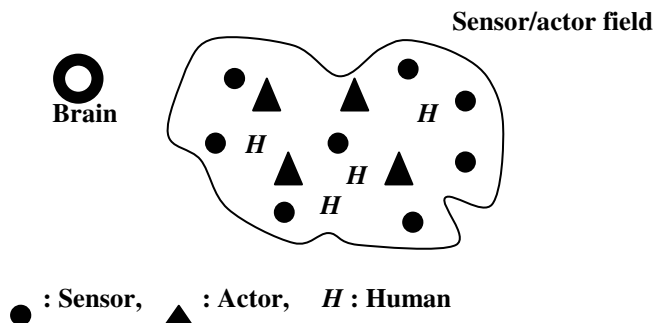
At present, the realization of (WSANs) is one of the challenging topics in the concerned research fields, and a considerable number of important issues have been proposed especially from the viewpoint of networking [8]-[10]. From the viewpoint of AI (Artificial Intelligence), a WSAN can be considered as an intelligent robot system with distributed sensors and actuators that can gather information of high density and perform appropriate actions upon its environment over wide areas.

The distributed intelligent robot network (DIRN) as shown in Fig.1, we propose here as one kind of WSAN, consists of one brain node and numerous sensor and actor nodes with human-friendly interfaces. It is assumed, for example, that sensors and actuators can collaborate autonomously to perform appropriate actions just like reflexive actions in humans and that the brain node works exclusively for complicated computation based on profound knowledge in order to control the other kinds of nodes, to communicate with people, etc.

In order to realize well-coordinated DIRNs, it is very important to develop a systematically computable knowledge representation language [11] as well as efficient networking technologies [10]. This type of language is indispensable to *knowledge-based* processing such as *understanding* sensory events, *planning* appropriate actions and *knowledgeable* communication even with humans, and therefore it needs to have

at least a good capability of representing spatio-temporal events that correspond to humans'/robots' sensations and actions in the real world.

Traditionally macro-commands such as 'move(10meters)' were employed for deploying sensors/motors. However, these commands were very specific to the devices and apt to have miscellaneous variants such as 'move(10meters, quickly)' and 'move(quickly, 10meters, leftward)', which is very inconvenient for communications especially between devices unknown to each other. Therefore, it is very important to develop such a language as is universal among all kinds of equipments.



**Fig. 1.** Physical architecture of DIRNs

Yokota, M. et al have proposed a semantic theory for natural languages so called 'Mental Image Directed Semantic Theory (MIDST)' [1], [2]. In MIDST, word concepts are associated with omnisensual mental images of the external or physical world and are formalized in an intermediate language  $L_{md}$  [11]. This language is employed for many-sorted first-order predicate logic with five types of terms. The most remarkable feature of  $L_{md}$  is its capability of formalizing both temporal and spatial event concepts on the level of human sensations while the other similar knowledge representation languages are designed to describe the logical relations among conceptual primitives such as words [3], [4].

The  $L_{md}$  was originally proposed for formalizing the natural semantics, that is, the semantics specific to humans, but it is general enough for the artificial semantics, that is, the semantics specific to each artificial device such as robot. This language has already been implemented on several types of computerized intelligent systems [1], [5], [11]-[13] and there is a feedback loop between them for their mutual refinement, unlike other similar ones [6], [7].

In this paper we focus on the semantic processing of sensory and action data represented in the formal language  $L_{md}$ , simulating the interactions between robots and their environments including humans.

## 2 A Brief Description of $L_m$

MIDST treats word meanings in association with mental images, not limited to visual but omnisensual, modeled as "Loci in Attribute Spaces". An attribute space corre-

sponds with a certain measuring instrument just like a barometer, a map measurer or so and the loci represent the movements of its indicator.

A general locus is to be articulated by “Atomic Locus” with the duration  $[t_i, t_f]$  and formalized as the expression (1). This is a formula in many-sorted first-order predicate logic, where “L” is a predicate constant with five types of terms: “Matter” (at ‘x’ and ‘y’), “Attribute Value” (at ‘p’ and ‘q’), “Attribute” (at ‘a’), “Event Type” (at ‘g’) and “Standard” (at ‘k’).

$$L(x,y,p,q,a,g,k) \quad (1)$$

The formula is called ‘Atomic Locus Formula’ whose first two arguments are sometimes referred to as ‘Event Causer (EC)’ and ‘Attribute Carrier (AC)’, respectively.

The intuitive interpretation of the expression (1) is given as follows, where ‘matter’ refers to ‘object’ or ‘event’.

*“Matter ‘x’ causes Attribute ‘a’ of Matter ‘y’ to keep ( $p=q$ ) or change ( $p \neq q$ ) its values temporally ( $g=Gt$ ) or spatially ( $g=Gs$ ) over a time-interval, where the values ‘p’ and ‘q’ are relative to the standard ‘k’.”*

When  $g=Gt$  and  $g=Gs$ , the locus indicates monotonous change or constancy of the attribute in time domain and that in space domain, respectively. The former is called a temporal event and the latter, a spatial event.

For example, the motion of the ‘bus’ represented by S1 is a temporal event and the ranging or extension of the ‘road’ by S2 is a spatial event whose meanings or concepts are formalized as (2) and (3), respectively, where the attribute is “physical location” denoted by ‘A12’.

(S1) The bus runs from Tokyo to Osaka.

$$(\exists x,y,k)L(x,y,Tokyo,Osaka,A12,Gt,k) \wedge bus(y) \quad (2)$$

(S2) The road runs from Tokyo to Osaka.

$$(\exists x,y,k)L(x,y,Tokyo,Osaka,A12,Gs,k) \wedge road(y) \quad (3)$$

MIDST has employed ‘tempo-logical’ connectives representing both logical and temporal relations between loci. A tempo-logical connective  $K_i$  is defined by (4), where  $\tau_i$ ,  $\chi$  and  $K$  refer to one of the temporal relations indexed by ‘i’, locus, and an ordinary binary logical connective such as the conjunctive ‘ $\wedge$ ’, respectively. This is more natural and economical than explicit indication of time intervals, considering that people do not consult chronometers all the time in their daily lives.

Here are introduced two examples of tempo-logical connectives, namely, ‘SAND’ and ‘CAND’. The expression (5) is the conceptual description of the English verb “fetch”, implying such a temporal event that ‘x’ goes for ‘y’ and then comes back with it, where ‘ $\Pi$ ’ and ‘ $\bullet$ ’ are instances of the tempo-logical connectives, ‘SAND’ and ‘CAND’, standing for “Simultaneous AND” and “Consecutive AND”, respectively. In general, a series of atomic locus formulas with such connectives is called ‘Locus formula’.

$$\chi_1 K_i \chi_2 \Leftrightarrow (\chi_1 K \chi_2) \wedge \tau_i(\chi_1, \chi_2) \quad (4)$$

$$(\exists x,y,p1,p2,k)L(x,y,p1,p2,A12,Gt,k) \bullet (L(x,y,p2,p1,A12,Gt,k) \Pi L(x,y,p2,p1,A12,Gt,k)) \wedge x \neq xy \wedge p1 \neq p2 \quad (5)$$

### 3 Specification of the World for a DIRN

‘The world for a DIRN’ ( $W$ ) refers to ‘the set of matters observable for the DIRN’ and is defined by (6) as the union of the set of its nodes ( $D$ ) and the set of the objects in its environment. The set  $D$  is the union of the sets of a brain node ( $\{B\}$ ), sensor nodes ( $Se$ ) and actor nodes ( $Ac$ ) as represented by (7) while the set  $O$  includes possibly humans and the other DIRNs.

$$W=D \cup O \quad (6)$$

$$D=\{B\} \cup Se \cup Ac \quad (7)$$

‘A constituent  $C_k$  of the world for a DIRN’ (i.e.,  $C_k \in W$ ) can be specified by the loci in the attribute spaces distinguishable by the sets of Attributes and Standards unique to the DIRN.

#### 3.1 Specification of Objects

An object in the environment of a DIRN (i.e.,  $C_k \in O$ ) can be characterized by the loci of its structure and so on. For example, the characteristics of a tree ‘C1’ in the environment can be represented by such a locus formula as (8), reading its height (A03) is between 4m and 5m, its location (A12) is in the park ‘C2’,...For another example, a road ‘C3’ that runs from a town ‘C4’ to a town ‘C5’ via a town ‘C6’ can be defined by (12), where ‘Me’ is the standard ‘Meter’.

$$\begin{aligned} \text{tree}(C1) \Leftrightarrow (\exists x, h, k, \dots) L(x, C1, h, h, A03, Gt, Me) \wedge (4m \leq h \leq 5m) \\ \wedge L(x, C1, C2, C2, A12, Gt, k) \wedge \text{park}(C2) \wedge \dots \end{aligned} \quad (8)$$

$$\begin{aligned} \text{road}(C3) \Leftrightarrow (\exists x, k, \dots) L(x, C3, C4, C6, A12, Gs, k) \bullet L(x, C3, C6, C5, A12, Gs, k) \\ \wedge \text{town}(C4) \wedge \text{town}(C5) \wedge \text{town}(C6) \end{aligned} \quad (9)$$

#### 3.2 Specification of a Sensor Node

A sensor node (i.e.,  $C_k \in Se$ ) can be specified by the loci of its structure and its collectable sensory data. In general, a sensor can be distinguished by the definition (10) from another kind of constituent where ‘data(y)’ is replaceable by (11), reading that a sensor ‘x’ is what takes in some data ‘y’ from some constituent ‘z’.

$$(\lambda x) \text{sensor}(x) \Leftrightarrow (\lambda x \exists y, z, g1, k1) L(x, y, z, x, p, q, A12, g1, k1) \wedge \text{data}(y) \quad (10)$$

$$(\exists z, z1, \dots, z_n, a, g, k, p1, \dots, p_{n+1}) (y=L(z1, z, p1, p2, a, g, k) \bullet \dots \bullet L(z_n, z, p_n, p_{n+1}, a, g, k)) \quad (11)$$

The formula (11) implies that ‘y’ is a locus in the attribute space referred to by the attribute ‘a’ and the standard ‘k’ unique to the sensor. For example, a thermometer ‘C7’ with the measurable range  $[-10^\circ\text{C}, +100^\circ\text{C}]$  can be characterized by (12) with the attribute ‘temperature (A28)’ and the standard of ‘Celsius (Ce)’.

$$\begin{aligned} (\exists y, z, g1, k, z1, \dots, z_n, p1, \dots, p_{n+1}) L(C7, y, z, C7, p, q, A12, g1, k1) \wedge \\ (y=L(z1, z, p1, p2, A28, Gt, Ce) \bullet \dots \bullet L(z_n, z, p_n, p_{n+1}, A28, Gt, Ce) \wedge (-10^\circ\text{C} \leq p1 \leq +100^\circ\text{C}) \\ \wedge \dots \wedge (-10^\circ\text{C} \leq p_{n+1} \leq +100^\circ\text{C})) \end{aligned} \quad (12)$$

### 3.3 Specification of an Actor Node

An actor (i.e.,  $C_k \in \mathcal{A}c$ ) can be specified by the loci of its structure, performable actions and, if any sensors with it, collectable sensory data. For example, a tanker ‘C8’ with the coverage [0km,100km] can be characterized by (13) with the attribute ‘mileage (A17)’.

$$(\exists x,r)L(C8,x,0,r,A17,Gt,Me) \wedge (0km \leq r \leq 100km) \wedge liquid(x) \quad (13)$$

### 3.4 Specification of the Brain Node

The brain node (i.e.,  $B$ ) can be specified by the loci of commonsense knowledge and the world knowledge including such specifications of the other constituents as mentioned above. For example, (14) is an example of commonsense knowledge, reading that ***a matter has never different values of an attribute at a time.***

$$L(x,y,p1,q1,a,g,k) \Pi L(z,y,p2,q2,a,g,k). \supset. p1=p2 \wedge q1=q2 \quad (14)$$

The intelligence of the brain node must be conscious of all about the other constituents but can be unconscious of the structure (e.g., hardware configuration) and the computational performance specification (e.g., CPU speed) of itself because they are what only meta-systems such as OS and meta-brain node have to concern. In our case, the brain node is a high-performance multimedia computer with the OS WINDOWS/XP and our intelligent system IMAGES-M installed [11].

## 4 Interaction Between a DIRN and Its World

The integrated multimedia understanding system IMAGES-M works as the main intelligence of the brain node of a DIRN. The intelligence of each sensor or actuator is a small-scaled IMAGES-M adapted for its specialized function. IMAGES-M has employed locus formulas as intermediate conceptual representations, through which it can integrally understand and generate sensor data, speech, visual image, text, and action data.

A DIRN is to solve some kinds of problems in its world. Such problems can be classified roughly into two categories as follows.

(CP) Creation Problem: e.g.) house building, food cooking, etc.

and

(MP) Maintenance Problem: e.g.) fire extinguishing, room cleaning. etc.

In general, an MP is relatively simple one that the DIRN can find and solve autonomously while a CP is relatively difficult one that is given to the DIRN, possibly, by humans and to be solved in cooperation with them.

### 4.1 Definition of a Problem and a Job for a DIRN

A DIRN must determine its job to solve a problem in the world. In general, the DIRN needs to interpolate some transit event  $X_T$  between the two events, namely, ‘Current Event ( $X_C$ )’ and ‘Goal Event ( $X_G$ )’ as shown by (15).

$$X_C \bullet X_T \bullet X_G \quad (15)$$

According to this formalization, a problem  $X_P$  is defined as  $X_T \bullet X_G$  and a job for the DIRN is defined as its realization.

The events in the world are described as loci in certain attribute spaces and a problem is to be detected by the unit of atomic locus by the inference employing such a postulate as (16) implying ‘Continuity in attribute values’. Therefore, the problem  $X_P$  in (17) is to be inferred as (18).

$$L(x,y,p1,p2,a,g,k) \bullet L(z,y,p3,p4,a,g,k) \rightarrow p3=p2 \quad (16)$$

$$L(x,y,p1,p2,a,g,k) \bullet X_P \bullet L(z,y,p3,p4,a,g,k) \quad (17)$$

$$L(z',y,p2,p3,a,g,k) \bullet L(z,y,p3,p4,a,g,k) \quad (18)$$

## 4.2 CP Finding and Solving

Consider a verbal command such as S3 uttered by a human. Its interpretation is given by (19) as the goal event  $X_G$ . If the current event  $X_C$  is given by (20), then (21) with the transit event  $X_T$  underlined can be inferred as the problem corresponding to S3.

(S3) Keep the temperature of ‘room C9’ at 20.

$$L(z,C9,20,20,A28,Gt,k) \wedge \text{room}(C9) \wedge (z \in \mathcal{O}) \quad (19)$$

$$L(x,C9,p,p,A28,Gt,k) \wedge \text{room}(C9) \quad (20)$$

$$\underline{L(z1,C9,p,20,A28,Gt,k)} \bullet L(z,C9,20,20,A28,Gt,k) \wedge \text{room}(C9) \wedge (z1 \in \mathcal{O}) \quad (21)$$

For this problem, the DIRN is to execute a job deploying a certain thermometer and actors ‘z1’ and ‘z’. The selection of the actor ‘z1’ is performed as follows:

*If 20-p < 0 then z1 is a cooler, otherwise  
if 20-p > 0 then z1 is a heater, otherwise  
20-p = 0 and no actor is deployed as z1.*

The selection of ‘z’ is a job in case of MP described in the next section.

## 4.3 MP Finding and Solving

In general, the goal event  $X_G$  for an MP is that for another CP such as S3 given possibly by humans and solved by the DIRN in advance. That is, the job in this case is to autonomously restore the goal event  $X_G$  created in advance to the current event  $X_C$  as shown in (22), where the transit event  $X_T$  is the reversal of such  $X_{-T}$  that has been already detected as ‘abnormal’ by the DIRN.

For example, if  $X_G$  is given by (19) in advance, then  $X_T$  is also represented as the underlined part of (21) while  $X_{-T}$  as (23). Therefore the job here is quite the same that was described in the previous section.

$$X_G \bullet X_{-T} \bullet X_C \bullet X_T \bullet X_G \quad (22)$$

$$L(z1,C9,20,p,A28,Gt,k) \wedge \text{room}(C9) \wedge (z1 \in \mathcal{O}) \quad (23)$$

## 5 Application to Robot Manipulation

The intelligent system IMAGES-M, still under development, is intended to facilitate integrated multimedia information understanding, including cross-media operations as shown in Fig.3. At present, IMAGES-M, installed on a personal computer, can deploy SONY AIBOs, dog-shaped robots, as actors and gather information about the physical world through their microphones, cameras and tactile sensors. Communications between IMAGES-M and humans are performed through the keyboard, mouse, microphone and multicolor TV monitor of the personal computer.

Consider such a verbal command as S4 uttered to the robot, SONY AIBO, named ‘John’.

(S4) John, walk forward and wave your left hand.

Firstly, late in the process of cross-media translation from text to AIBO’s action, this command is to be interpreted into (24) with the attribute ‘shape (*A11*)’ and the values ‘*Walkf-1*’ and so on at the standard of ‘*AIBO*’, reading that John makes himself walk forward and wave his left hand. Each action in AIBOs is defined as an ordered set of shapes (i.e., time-sequenced snapshots of the action) corresponding uniquely with the positions of their actuators determined by the rotations of the joints. For example, the actions ‘walking forward (*Walkf*)’ and ‘waving left hand (*Wavelh*)’ are defined as (25) and (26), respectively.

$$\begin{aligned}
 &L(\text{John,John,Walkf-1,Walkf-m,A11,Gt,AIBO}) \\
 &\wedge L(\text{John,John,Wavelh-1,Wavelh-n,A11,Gt,AIBO}) \quad (24)
 \end{aligned}$$

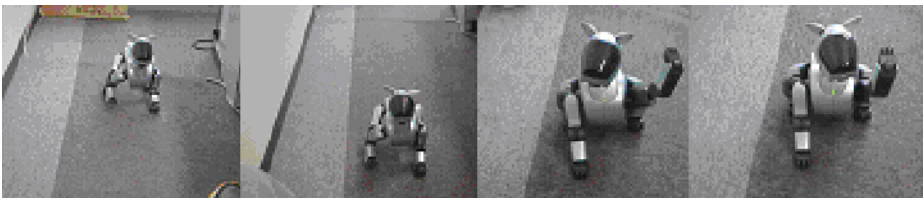
$$\text{Walkf}=\{\text{Walkf-1, Walkf-2,...,Walkf-m}\} \quad (25)$$

$$\text{Wavelh}=\{\text{Wavelh-1, Wavelh-2,..., Wavelh-n}\} \quad (26)$$

Secondly, an AIBO cannot perform the two events (i.e., actions) simultaneously and therefore the transit event between them is to be inferred as the underlined part of (27) which is the goal event here.

$$\begin{aligned}
 &L(\text{John,John,Walkf-1,Walkf-m,A11,Gt,AIBO}) \\
 &\bullet \underline{L(\text{John,John,Walkf-m,Wavelh-1,A11,Gt,AIBO})} \bullet \\
 &L(\text{John,John,Wavelh-1,Wavelh-n,A11,Gt,AIBO}) \quad (27)
 \end{aligned}$$

Thirdly, (28) is to be inferred, where the transit event, underlined, is interpolated between the current event and the goal event  $X_G (= (27))$ .



**Fig. 2.** AIBO (Sony) behaving in accordance to the command ‘Walk forward and wave your left hand’



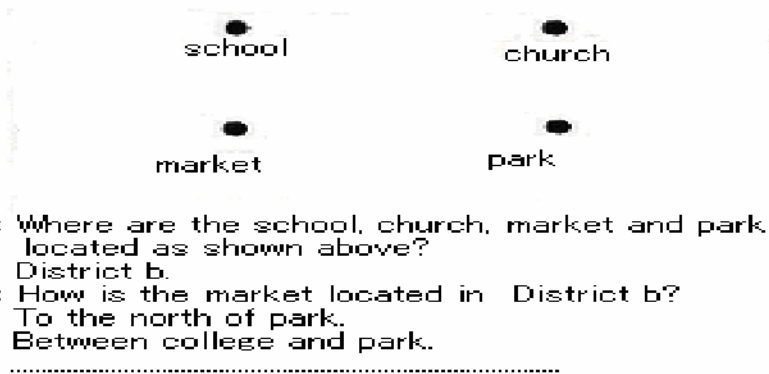
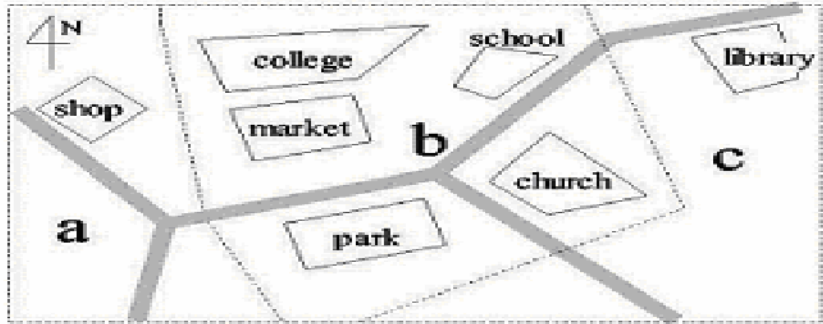


Fig. 3. Example of cross-media Q-A between humans (H) and IMAGES-M (S)

$$L(\text{John,John,p1,p2,A11,Gt,AIBO}) \\ \bullet L(\text{John,John,p2,Walkf-1,A11,Gt,AIBO}) \bullet X_G \quad (28)$$

Finally, (28) is interpreted into a series of joint rotations in the AIBO as shown in Fig.2.

## 6 Discussion and Conclusion

As the simulations of DIRN-world interactions, several kinds of cross-media operations via locus formulas have been tried. At our best knowledge, there is no other system that can perform cross-media operations in such a seamless way as ours [11]. This leads to the conclusion that employment of atomic locus formulas has made the logical expressions of event concepts remarkably computable and has proved to be very adequate to systematize cross-media operations. This is due to their medium-freeness and good correspondence with the performances of miscellaneous devices, which in turn implies that locus formula representation may make it easier for the devices to share a task than macro-command representation.

From the simulation results, we conclude that  $L_{md}$  can be a universal language for WSANs including DIRNs. Our future work will include establishment of learning facilities for automatic acquisition of word concepts from sensory data and human-robot communication by natural language under real environments.

## Acknowledgements

This work was partially funded by the Grants from Computer Science Laboratory, Fukuoka Institute of Technology and Ministry of Education, Culture, Sports, Science and Technology, Japanese Government, numbered 14580436 and 17500132.

## References

1. Yokota,M.: An approach to natural language understanding based on a mental image model. Proc. of the 2<sup>nd</sup> International Workshop on Natural Language Understanding and Cognitive Science (2005) 22-31
2. Yokota,M. et al: Mental-image directed semantic theory and its application to natural language understanding systems. Proc. of NLPRS'91(1991) 280-287
3. Sowa,J.F.: Knowledge Representation: Logical, Philosophical, and Computational Foundations. Brooks Cole Publishing Co., Pacific Grove, CA, (2000)
4. Zarri,G.P.: NKRL, a Knowledge Representation Tool for Encoding the 'Meaning' of Complex Narrative Texts. Natural Language Engineering - Special Issue on Knowledge Representation for Natural Language Processing in Implemented Systems, 3 (1997) 231-253
5. Oda,S., Oda,M., Yokota,M. : Conceptual Analysis Description of Words for Color and Lightness for Grounding them on Sensory Data. Trans. of JSAI,Vol.16-5-E (2001) 436-444
6. Langacker,R.: Concept, Image and Symbol, Mouton de Gruyter, Berlin/New York (1991)
7. Miller,G.A., Johnson-Laird,P.N.: Language and Perception, Harvard University Press (1976)
8. Akyildiz,I.F., Su,W., Sankarasubramaniam,Y., Cayirci,E: Wireless sensor networks: a survey. Computer Networks, 38-4 (2002) 393-422
9. Haenggi,M.: Mobile Sensor-Actuator Networks: Opportunities and Challenges. Proc. of 7th IEEE Int. Workshop, Frankfurt,Germany (2002)283-290
10. Akyildiz,I.F., Kasimoglu,I.H.: Wireless Sensor and Actor Networks: Research Challenges. Ad Hoc Networks, 2 (2004)351-367
11. Yokota,M., Capi,G.: Cross-media Operations between Text and Picture Based on Mental Image Directed Semantic Theory, WSEAS Transactions on Information Science and Applications, 10-2 (2005) 1541-1550
12. Amano,M., et al : Linguistic interpretation of human motion based on Mental Image Directed Semantic Theory, Proc. of IEEE AINA-2005, Taipei (2005) 139-144
13. Amano,M., et al : Cross-media translation between human motions and texts based on Mental Image Directed Semantic Theory, Proc. of IEEE ICDCSworkshop-2005, Ohio (2005) 707-713

# Hyper-Interactive Video Browsing by a Remote Controller and Hand Gestures

Hui-Huang Hsu, Timothy K. Shih, Han-Bin Chang,  
Yi-Chun Liao, and Chia-Tong Tang

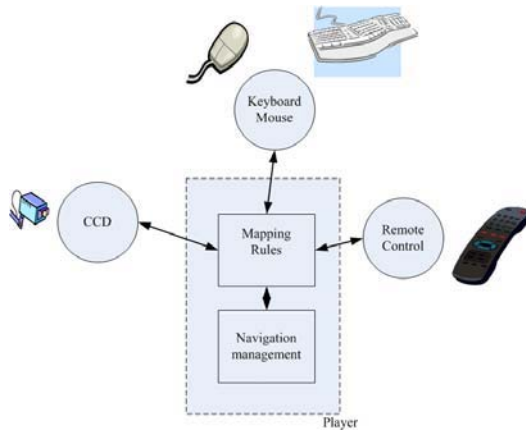
Multimedia Information Network Lab,  
Department of Computer Science and Information Engineering,  
Tamkang University, Taiwan, R.O.C.  
{hhsu, tshih}@cs.tku.edu.tw

**Abstract.** Interactive video browsing tools are designed for e-learning applications on future interactive TVs. The integrated system includes an authoring tool that produces multi-paths videos and a playback tool that uses video tracking technology and a remote controller. The playback tool enables multi-modal interaction between the user and a multi-story video clip. Three types of hyper-interactive controls are incorporated, which include a reference link of a video object to show supplementary information on the Web, a hyper link to enable hyper-video jumps, and a choice link for online answers to pre-designed questions. The underlying video is coded using the standard MPEG technology, with navigation information hidden in the user-defined data of MPEG. Thus, the default sequence of a hypervideo can also be presented using an ordinary video player.

## 1 Introduction

Interactive TV aims at giving full interactivity between the TV programs and the viewers. The viewers will be able to select needed video in a simple and easy way. The interface design and related issues has been under massive research and discussions in recent years [1, 2, 3]. On the other hand, hypervideo and multi-story video are also an important video technology under development [4, 5, 6, 7]. Video not only can be played in a linear sequence, but also can have multiple choices at certain points in the video. The story of a movie can progress in different ways and have different endings chosen by the viewer. An object in the video can be annotated with extra information in the form of a text file, an image, another clip of video, or a Web page. As long as the viewer triggers the reference link associated with it, the information will reveal. It would be very useful to integrate the hypervideo technology into interactive TVs to further enhance the interactivity.

How the reference link in a video can be triggered is an interesting issue. It is quite natural to do it by clicking a mouse, just like what is usually done on the World Wide Web. However, a mouse seems not well suited in the living room. People are more used to play with the buttons on a remote controller. In this paper, we propose multi-modal interaction for the playback tool. Besides using a remote controller, hand gesture of the viewer captured by a CCD camera can also be used for controlling hyper-video progress. The idea is shown in Fig. 1.



**Fig. 1.** Multi-modal hyper-interactive video viewing

For an e-learning application in the form of video compact disc, the learner will not need a computer to view the hypertext content. A DVD player and a TV, which are standard appliances in the living room of a normal family, would be sufficient for the learner to start learning. Personalized learning content can be retrieved through the hypertext structure of the learning content. For example, the learner can choose to or not to view a video clip with detail explanation of a vocabulary in a conversation for a language-learning scenario. Computers will not be needed.

In the following sections, we will introduce the authoring tool and the player of this integrated system. The multi-modal interaction and enabling technologies will then be delineated. And a brief conclusion will be drawn.

## 2 Hyper-Interactive Video Content Authoring

In order to produce hyper-interactive video content, an authoring tool is developed to divide the original video raw file into video clips. A directed-graph (digraph) structure composed of the video clips can then be constructed. One video clip can be used more than once in the digraph structure. The content producer can use the authoring tool with a user-friendly interface to produce the clips by simply marking in the starting time and marking out the ending time. Actually, the video file is not edited in any way. It is the marked points that are saved.

Fig. 2 shows the appearance of our authoring tool. There are several windows in this tool, including the video window, a digraph structure of marked hypertext, and the metadata that can be added into the video.

The basic element of hypertext is a video clip. A hypertext is composed of several video clips. In our digraph-structured hypertext, every node represents a video clip marked by the producer. The root node is the starting point of the hypertext and is marked with a red square in the authoring tool. The red line between two nodes represents a video hyperlink between the two video clips. The audience can activate the hyperlink in a specified temporal-spatial domain to jump from one video clip to another. Nodes can have more than one branch. So the user can decide which video

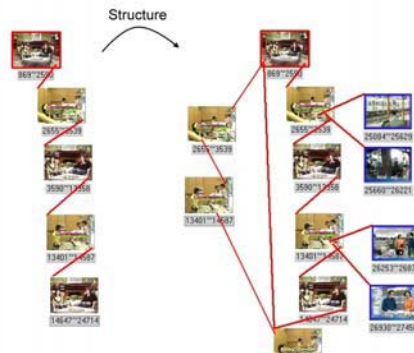
clip is the target via the multi-modal interaction. One of the hyperlinks of each node is set as the default link. If the audience does not make any choice, the video playing will proceed to the video clip with the default link.



**Fig. 2.** The authoring tool of interactive-hyper video

Fig. 3 shows the result of a hypervideo structure. The producer can change the linearly played video sequence into a digraph-structured hypervideo. The audience can select the video clip he/she wants to watch by triggering a reference link. If they do not like the selected clip, they can go back to the parent video clip to choose another video hyperlink. The producer can also put some extra information to describe certain objects in the video clip. By the authoring tool, the producer can add text descriptions, existing image files, webpage files or URLs on the Internet to give more information to the audience.

Video annotation is usually used to enhance the semantics of the video object in the research of MPEG-4, MPEG-7 and video retrieval [8]. It is a big challenge to decide which objects to be recognized, tracked, and annotated. In our work, we adopt a manual and intuitive way to reduce the complexity of authoring.



**Fig. 3.** A linear sequence versus a directed graph-like structure

### 3 The Hypervideo Player

In the proposed system, one key issue is how the annotated video can be played in a selected sequence exactly. A hypervideo presentation engine is designed for the hypervideo player. There are three components in the presentation engine: the navigation manager, the video decoder, and the video render (Fig. 4). They are described in details in the following.

- Navigation\_Manager:**  
 The major component of the presentation engine is the navigation manager that lets the user browse the video sequence and receives the multi-modal interaction signal from the user. It can also control the process of video decoding when the user jumps to the next and previous video clips.
- Video\_Decoder:**  
 This component is responsible for decoding the video signal like an MPEG decoder. The video decoder installed in the Windows OS is used directly.
- Video\_Renderer:**  
 This component is used to render a video that comes from the output of the navigation manager and the video decoder. If one of the inputs to the video render is interrupted, the video render will output the default video sequence without using the hypervideo function.

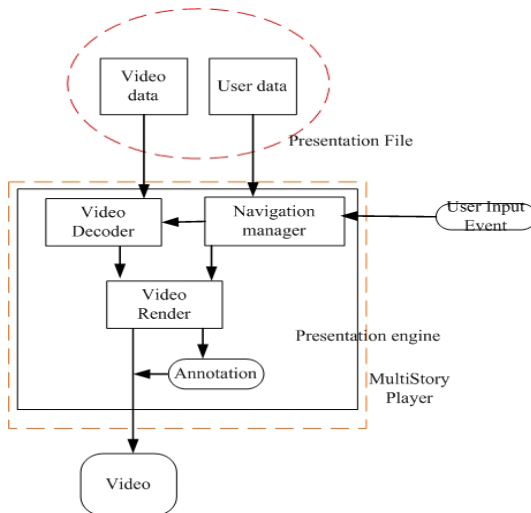


Fig. 4. The architecture of the hypervideo player

The video produced by the hypervideo authoring tool can also be played in other video players. Other players will play the default sequence of the digraph structure. The annotation added by the authoring tool will be ignored.

## 4 Multi-modal Interaction

Mouse clicking is natural to computer users on the WWW environment. Hypervideo is an extension of the idea of hypertext in Web pages to video. But here we look for other ways of interactions for future interactive TV applications. Under the scenario, people might not be used to mouse clicking. Thus, two other modes of interactions are introduced: 1. interaction with a remote controller, 2. interaction with camera-captured hand gestures.

### 4.1 Interaction with a Remote Controller

The first recommended device is a remote controller for TV. Most people are used to pressing buttons of a remote controller. However, current design of a TV remote controller needs to be enhanced to incorporate certain functions of the hypervideo player. A few buttons are redefined for such a purpose. The workflow of the remote control is shown in Fig. 5. In order to simulate the TV on a computer screen, an IR (infrared)

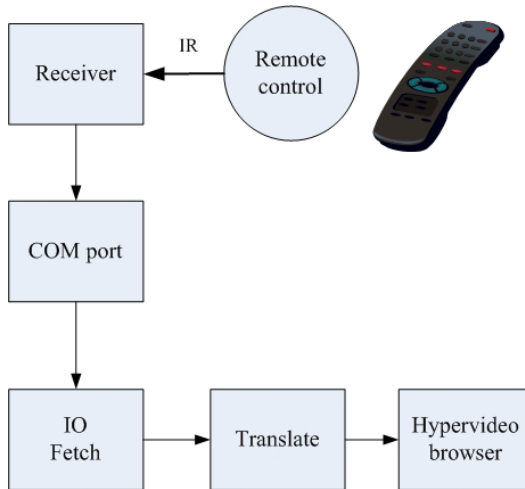


Fig. 5. The workflow of the remote control for the hypervideo player



Fig. 6. The IR receiver and the remote controller

rays) receiver is added to the COM port to receive the IR signal from the remote controller. The IO fetch component receives data from the COM port and translates the data for the hypervideo player. Fig. 6 shows the IR receiver and the redefined remote controller.

Fig. 7 shows the situation that a viewer uses a remote controller to browse a hypervideo. The hyperlinks are shown in the video window with numbers. The viewer can activate the text, image, Web page, and/or video hyperlinks associated with the video clips by simply pressing the number buttons. Besides the hyperlinks, the viewer can also switch between play and pause or jump to a certain video clip.



**Fig. 7.** Using a remote controller to browse hypervideo

## 4.2 Interaction with Camera-Captured Hand Gestures

A CCD camera is used as the second input device for the viewer to interact with the hypervideo browser. Hand gestures are used to replace mouse events. When the viewer waves his/her hand, the system finds the center of the palm and it is viewed as the mouse cursor. The viewer moves the hand to move the cursor to a certain hyperlink. When the palm is folded into a fist, the hyperlink is triggered. A low level CCD is sufficient to capture the user's gestures. Such a device can be acquired easily at a low cost.

Gestures are used as the event to trigger a hyperlink. The first step of recognizing gestures is to separate the hand in each frame. The second step is to find the center of gravity of the palm as the location of the mouse cursor. And the third step is to detect the palm is folding or unfolding [9, 10, 11].

The following procedures are used to separate the hand from the background:

1. Compute the difference of lightness in each pixel between the current frame and the former frame.
2. If the result is greater than the threshold, the value is set to 255. Otherwise, the value is set to 0.

This can be expressed by the following equation.

$$P_j(x, y) = \begin{cases} 1 & \text{if } |I_j(x, y) - B_j(x, y)| > T \\ 0 & \text{Otherwise} \end{cases}$$

With an adequate threshold, we can get results of separating the foreground object from a still background.

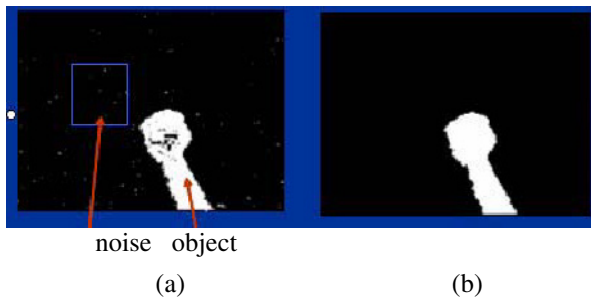


Next we need to reduce the noise shown with the foreground object. The median filter and the closing operation are used. Here, the two techniques are briefly introduced.

**Median filter:** All the pixels values (in gray-level) in an  $N*N$  mask are arranged in a sequential order (from the smallest to the largest), then the middle value is selected from the ordered set to replace the value of the central pixel.

**Closing operation:** The closing operation performs the dilation followed by the erosion operation. Usually, it is used to fill in small holes or gaps and connect object's fragments.

By using these techniques, we can remove the redundant noise and get the hand area more precisely. Fig. 8 shows the hand area after removing noise.



**Fig. 8.** Result of noise reduction (a) before and (b) after the noise reduction

Illumination change and background change are the two factors for getting an inaccurate background image. Without an accurate background, the foreground object cannot be separated. So, a dynamic background update mechanism is necessary. The major component of background adaptation is to calculate the difference between the current frame and background image. If the difference exceeds a threshold, the background image must be updated.

In this system, we use the number of on-off times of each hand object to determine if it is folding or unfolding. The procedure of determining on-off times is as follow:

1. Scan the source image horizontally from the top to the bottom with an interval of two pixels until whole image is scanned.
2. In the scanning process, the system checks the value change of neighboring pixels.
3. If there is a change from black to white, it is an "on."
4. If there is a change from white to black, it is an "off."
5. Sum up the total number of on's and off's.

With an adequate threshold, we can use the on-off number to determine whether the palm is folding or unfolding. Fig. 9 shows the two gesture configurations.

In our system, hand gestures are used to trigger the hyperlink in a hypervideo. The center of the palm is considered as the cursor to locate the target hyperlink. After the above-mentioned procedure, we can easily get the area of the hand. Then we compute the center of gravity of the palm to get a coordinate and use this position to select a hyperlink. Fig. 10 shows a locating result using the center of gravity of palm.

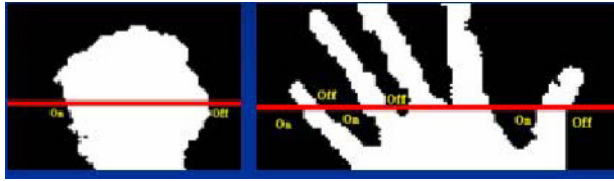


Fig. 9. Gestures: folding and unfolding



Fig. 10. Mouse moving and clicking by gestures

When the viewer moves his/her hand, the captured frame is analyzed and a coordinate is returned to the hypervideo player. The hypervideo render engine renders a small block indicating the location the palm points to. When the viewer folds the palm, the hypervideo player detects a low on-off number. This action is treated as the event of mouse clicking. So if there is a hyperlink in the area, it will be activated by the action.

## 5 Conclusions

In this research we integrated the hypervideo structure into the interactive TV framework. With the developed multi-modal interaction tools, the viewer can sit back and relax on a couch to learn an e-learning application with hypervideo content and enjoy the interactivity easily. The interactivity of the interactive TV is enhanced with the hypervideo.

## Acknowledgement

This work was partially supported by National Science Council, R.O.C. under grant number NSC 94-2213-E-032-018.

## References

1. Bing J., Dubreuil J., Espanol J., Julia L., Lee M., Loyer M., Serghine M., "MiTV: rethinking interactive TV," in *Proceedings Seventh International Conference on Virtual Systems and Multimedia* Oct. 25-27, 2001
2. Liang-Jie Zhang, Jen-Yao Chung, Lurng-Kuo Liu, Lipscomb J.S., Qun Zhou, "An integrated live interactive content insertion system for digital TV commerce," in *Proceedings Fourth International Symposium on Multimedia Software Engineering*, Dec. 11-13, 2002.
3. Cesar P., Vierinen J., Vuorimaa P., "Open graphical framework for interactive TV," in *Proceedings Fifth International Symposium on Multimedia Software Engineering*, Dec. 10-12, 2003.
4. Chang, H.-B., H.-H. Hsu, Y.-C. Liao, T. K. Shih, and C.-T Tang, "An Object-Based HyperVideo Authoring System," in *CD-ROM Proceedings of the Int'l Conf. on Multimedia Expo*, June 28-30, 2004.
5. Yoshiaki Hada, Hiroaki Ogata, and Yoneo Yano, "XML-based Video Annotation system for Language Learning Environment," in *Proceedings of the Second International Conference on Web Information Systems Engineering*, Vol. 1, Dec. 3-6, 2001.
6. Correia, P.L. and Pereira, F., "Objective evaluation of video segmentation quality," in *IEEE Transactions on Image Processing*, Vol. 12, Issue 2, Feb. 2003.
7. Nitin Sawhney, David Balcom, and Ian Smith, "Authoring and navigating video in space and time," in *IEEE Multimedia*, Volume 4, Issue 4, Oct.-Dec. 1997.
8. Jim Taylor, *DVD Demystified* (2nd edition), McGraw-Hill Professional, Dec. 2000.
9. Oka, K., Sato, Y., and Koike, H., "Real-time tracking of multiple fingertips and gesture recognition for augmented desk interface systems," in *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, May 20-21, 2002
10. Yang Liu and Yunde Jia, "A robust hand tracking and gesture recognition method for wearablevisual interfaces and its applications," in *Proceedings of the Third International Conference on Image and Graphics*, Dec. 18-20, 2004.
11. Dias, J.M.S., Nande, P., Barata, N., and Correia, A., "OGRE - open gestures recognition engine," *Proceedings of the 17<sup>th</sup> Brazilian Symposium on Computer Graphics and Image Processing*, Oct. 17-20, 2004.

# Mobile Computing with MPEG-21

Marios C. Angelides<sup>1</sup>, Anastasis A. Sofokleous<sup>1</sup>, and Christos N. Schizas<sup>2</sup>

<sup>1</sup> Brunel University, Uxbridge, Middlesex, UB8 3PH, UK  
{marios.angelides, anastasis.sofokleous}@brunel.ac.uk

<sup>2</sup> University of Cyprus, Nicosia, Cyprus  
schizas@ucy.ac.cy

**Abstract.** In the field of multimedia content provision, many researchers investigate numerous techniques for adapting the content based on device, network, environment and user characteristics. In this paper, we present an MPEG-21 middleware design for dynamic content adaptation and we demonstrate a true mobile multimedia client that is able to communicate information about device and network characteristics and provide the necessary tools for the configuration of user preferences.

## 1 Introduction

MPEG-21 aims at understanding how the various elements of the infrastructure for the deployment of multimedia applications fit together. Various terms and definitions have been defined in this framework, such as content which have been replaced by the term Digital Item. Digital item adaptation is defining the personalization of the selection and delivery of multimedia content to the individual users. Content adaptation is necessary so as to make the content universally accessible (Universal Multimedia Access) since different networks and devices imply different quality of service, bit rate, computing and presentation capabilities and also different users imply different preferences (e.g. content type, content quality, etc) and usage history[1],[2]. However, MPEG-21 provides only some of the tools necessary to support resource adaptation, descriptor adaptation and QoS management.

Not for all the content types the adaptation operations are the same. For instance, while video adaptation include operations such as transcoding, video summarization, replacement, synthesis [3], for audio the operations are specified as *channelDropping* and *scalableAudio* [4]. Using each one of these operations, a digital item will provide different results which will certainly have different satisfaction for each user based on his/her preference and his/her current activities or location. However, the objective usually is not just to adapt the digital item but to provide a result which will be optimized for the network, fit to device characteristics but also have the best satisfaction for the end-user. Therefore the selection of either a single or a combination of operations requires a number of algorithms for determining the operation and adapting the digital item and evaluating the result quality. While numerous types of terminal and mobile devices are being used for content retrieval, currently users' concerns lie not with the end-system, but with the information and information quality[5].

In this paper, we present how the MPEG-21 framework will assist in content adaptation and we propose an MPEG-21 middleware design. In addition we present a true multimedia client for mobile devices that uses the current MPEG-21 framework for acquiring the dynamically adapted content. Section 2 discusses the research overview in this area while section 3 presents the middleware design and illustrates the multimedia mobile client. Finally, the paper concludes and presents future work.

## 2 Research Overview

During dynamic adaptation, a system, which is responsible for the whole processing, has to reflect on the adaptation operation characteristics and also the content, user, environment, network and device characteristics. In addition, the delivery of the content, which is mostly depended on the adaptation result, invokes the Quality of Service, a value that is not easy to determine automatically. Having this in mind, it is obvious why until now, in most of the approaches, content adaptation process is a communication handshake calling for a user to select most of the attributes (e.g. quality, format of the content) from an inadequate predefined number of choices. A concise outline of the current research work is discussed below.

In [6], a three-tier architecture was implemented for selecting, adapting and delivering personalized and summarized content to the end users. The media middleware consists of the personalization engine and adaptation engine. In this work, personalization is focused on the insertion or deletion of each video shot depending on user preferences. It is not quite clear how the adaptation engine is achieving the optimal adaptation, but as it is cited, part of the algorithm extracts audiovisual segments from shot boundaries and combines them based on the personalization selection. Moving from ad-hoc content adaptation to a most intelligent adaptation, it is necessary to define the relations of video entity, adaptation, resource and utility [7]. In [7] as in [3] as well, the authors used the term “*adaptation operation, resource and utility (ARU) spaces*” in order to emphasize the multidimensional constrained problem during the adaptation process. To be more precise, given a particular content unit, adaptation operation space relates to the existence of many adaptation methods that can be applied, such as transcoding, summarization, etc. Furthermore, resource space is defined by the multiplicity of the device and network characteristics related with the content delivery and consumption, such as bandwidth, device computational capabilities, display dimensions, etc. The utility space is quite nearer to the quality of service concept, as it is measuring in its multiple dimension space the user’s likings and preferences. Therefore, a point at adaptation space (e.g. corresponded by a defined adaptation operation such transcoding) is associated with’ specific resources and utility values which are represented by corresponding points in the resource space and the utility space respectively. Although the above seem to formulate a complete framework for dynamic adaptation, however, during dynamic adaptation, the main problem lies with the fact that the utility value can not be easily measured. Therefore, for a given resource-constrained utility maximization optimization problem, the result of the adapted content can not be easily evaluated without the human factor. In addition, during the searching for the appropriate adaptation operation and its parameters, it is very difficult to know the result of each selection without exercising it at that point of time.

### 3 Design Overview

The collection, preparation, delivery and consumption chain of content (or digital item) is very long and ambiguous process that begins with the author and usually ends up at the user. A number of actors are involved in the process such as the author, the publisher, the content provider, the content consumer. In addition, many factors and parameters affect that chain, such as the network environment, natural environment, device and user. Whatever device is used (e.g. a normal PC, a mobile phone, a pocket pc, a handheld device, etc) the content must be able to plug-and-play on that device. Therefore, the specific device characteristics must be collected and send for tuning the content according to those characteristics. Usually to the responsible party for this kind of processing is middleware.

Applications supporting MPEG-21 consist of many application layers. Each layer is responsible for a specific functionality, and communication among the layers is achieved through exchange of messages between neighbor layers. For instance, validation for the XML schema or documents would be made by an XML parser. Basic functionality, that is required for each common MPEG-21 tool include mechanisms for search, updating and insertion. Using the basic functionality, the business process may be customized for each application based on its specific requirements. Finally, an application (graphical or console) user interface may be placed on top of the architecture, e.g. a customer user interface for the client, an administration interface for the server, or an application interface providing connection with external systems. The MPEG-21 middleware involves the interaction of many crucial processes, algorithms and operators (e.g. selection[9], adaptation process, adaptation operators, database utilities, content analysis, content evaluation process) but also requires the involvement of MPEG-21 schema and xml readers & writers. The MPEG-21 framework proposes that during digital item consumption, content adaptation can be utilized through a resource adaptation engine and/or a descriptor adaptation engine [10], [11].

Figure 1 illustrates a likely design of the content adaptation process. In a typical scenario, a user requests a multimedia object (digital item). In the particular case scenario, the digital item is MPEG-7 defined, thus we assume that the annotation step has already taken place. A system (e.g. content provider system) should offer the item to the user in the appropriate format. Specifically, the item has to be adapted for fitting into device, network and environment characteristics, such as terminal display's dimension, bandwidth limitations, etc. While, the information of the digital item is retrieved from the MPEG-21 schema, the information for the terminal and user's preferences is retrieved in a MPEG-21 format from a storage location (e.g. profile storage manager). A number of available adaptation operations assist in the adaptation process. However, the main issue here is what adaptation operation is more suitable to be used and in addition what parameters should be utilized with the particular operation. For that reason, a selection process selects and subsequently tunes the most appropriate adaptation operation with the aid of utilities and several types of information. Among these utilities are modules responsible for content and usage analysis and user-centered generated content evaluation.

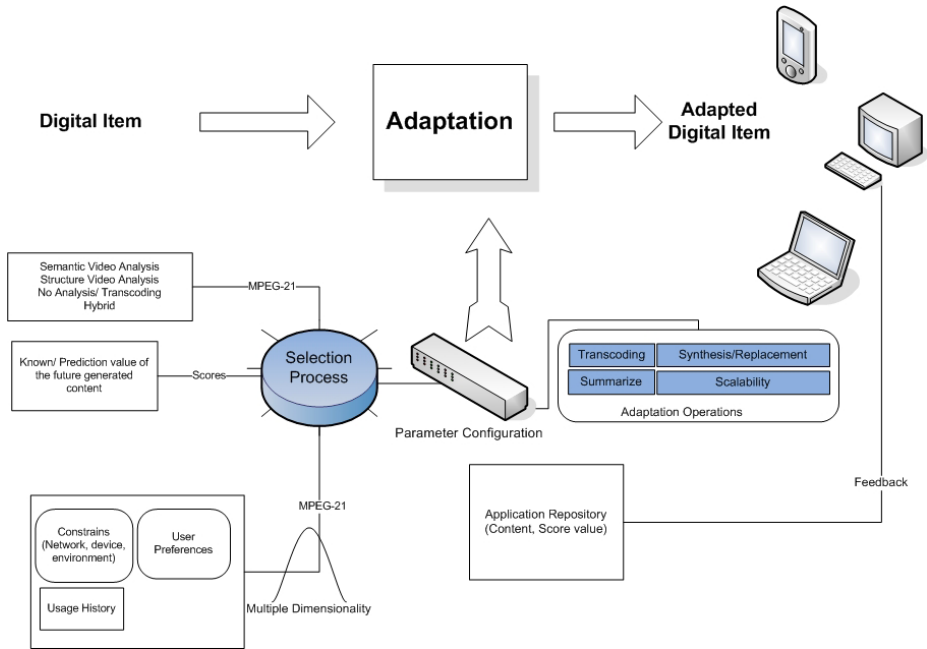


Fig. 1. MPEG-21 Design

Dynamic digital adaptation, which is a module of the middleware and will mostly include optimization algorithms, is designed to interface with the top layer of the MPEG-21 middleware in order to request, adapt, store and provide a digital item along with its descriptors. MPEG-21 middleware is currently under development based on the above design. The following section discusses implementation of the MPEG-21 middleware.

### 3.1 The MPEG-21 Middleware Implementation

The multimedia content adaptation under the MPEG-21 framework is utilized after the digital item request. The *Digital Item Upload* is part of the content management process followed by each author or publisher. During that phase, a video accompanied by a MPEG-7 file is uploaded to the server. The MPEG-7 file describes the video file syntactically and semantically. For instance, a sport video clip, being stored in MPEG-4 video format, illustrates a soccer player kicking the ball. Therefore, the MPEG-7 file includes the video format characteristics, the description about this event (e.g. date, time, occasion, location, player name, and playing field name), description about the objects of this video clip and their actions, etc. We have developed a web user interface connected with the middleware. Through that, a user can upload digital items (e.g. a video, an image, a music file) with the MPEG-7 file. The MPEG-7 is linked with the digital item and stored in a database.

On the other side, a user requests to watch the particular video clip using his/her mobile device (*Digital Item Request*). However, while he is in a hurry, he requests

that he would see the video in black and white color. In addition his device only supports MPEG-2 video and is connected into the internet with GPRS over GSM. Therefore, a MPEG-21 file (constructed at the user site) is sent to the middleware providing the current characteristics of the particular transaction (e.g. his preferences, his devices characteristics, and the network characteristics). The multimedia mobile client is explained in the next section.

The middleware receives the MPEG-21 file(s) and using this file as a parameter calls a selection process. The selection process loads all the information about the user using the MPEG-21 files and the already stored information. User's MPEG-21 characteristics along with his/her usage history data are loaded. The selection process aims to identify the appropriate adaptation operation and therefore to make an appropriate decision on how to adapt an input bit-stream based on many input constrains and characteristics. This kind of decision-taking is characterized as a constrained optimization problem involving algebraic variables that represent adaptation parameters, media characteristics, usage environment inputs, or any combinations thereof [9]. The solution, yielding the decision, can then be computed by a universal process independent of what the variables represent. Analyzing the MPEG-7 file and the MPEG-21 file, the selection process has to search for an adaptation operation that will transform the MPEG-4 color video to the MPEG-2 black/white video. The transcoding operation is selected and is utilized (*Digital Item Adaptation*).

The middleware is streaming the on-air adapted digital item (*Digital item provision*). The mobile device receives the streaming video and display the black/white video to the user (*Digital item consuming*). The user can evaluate the content using his device browser (explained also in the next section). End-user devices must be able to interact with the middleware infrastructure so as to provide the necessary information (e.g. characteristics and user' preferences). The interaction will guide the adaptation process. In addition a digital item viewer is needed fore viewing adapted items. The following section presents a mobile application, called MPEG-21 mobile client, which consists of an interactive MPEG-21 browser and a digital item viewer.

### 3.2 MPEG-21 Mobile Client

The proposed system consists of a digital item Viewer, a sub-system mainly designed and developed for mobile devices, and a MPEG-21 interactive browser (figure 2). Both of the sub-systems are implemented in J2ME (java for mobile devices) and a number of extra J2ME libraries which come with mobile phones (e.g. MMPAPI). Thus, the MPEG-21 mobile client is able to run on any mobile device. Using this integrated and complete mobile system, a user may specify his/her preferences under the MPEG-21 framework and therefore configure digital items according to User preferences[12], request adaptation for resources, and view the desired digital item adapted by the MPEG-21 middleware.

*MPEG-21 Interactive Browser* communicates directly with the MPEG-21 middleware and provides the necessary tools and specifications for both user and device total interaction under the MPEG-21 standard. The expandable browser comprising many components, enhances the optimization of the content adaptation process, since during communication with the middleware, it automatically provides information about the mobile device and network characteristics.



Many user preferences (e.g. content, video and audio presentation, color, presentation priority, etc), which also are proposed by the MPEG-21 experts, shape a user profile and therefore guide content adaptation. When a person uses a different mobile device or terminal, his/her preferences may not change. That kind of action requires having user's profile portable, which implies either a number of algorithms for moving intelligently the profile each time a user change a device or just to have each user's profile at a central online server. In addition, with regards to customizing preferences for each device, there will be few differences to opting for a fresh specification. The user profile may be saved in repositories[13] or in databases[14]. In DIA, session mobility refers to the transfer of configuration - state information that pertains to the consumption of a Digital Item on one device to a second device[15].

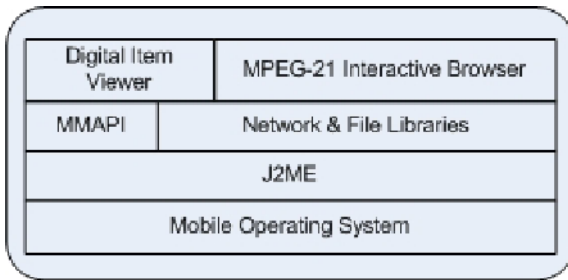


Fig. 2. The Multimedia Mobile Client

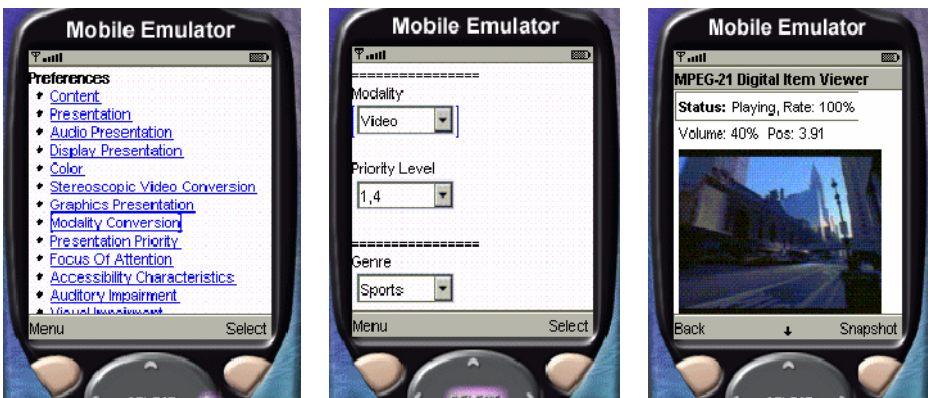


Fig. 3. (a) MPEG-21 User Preferences. (b) Modality Priority. (c) MPEG-21 Digital Item Viewer.

The MPEG-21 interactive browser identifies the user, and loads his/her preferences during initialization. Furthermore, a user is able to update his/her preferences at any time (figure 3a). User characteristics include general user information, content preferences, presentation preferences, accessibility characteristics, mobility characteristics and destination [16]. Currently most of these have to be inserted manually since most of the preferences cannot be extracted anywhere else but from

the user. User location can be determined and inserted automatically by the system. The interactive browser can be easily expanded and modified to support location services so as to provide the location information to MPEG-21 middleware. Therefore, adaptation algorithms will use this kind of information in their calculations for better digital item adaptation since specific preferences usually vary based on location. For instance, users would like to watch high quality movies at home even when there is a request to watch in black and white rather than color.

Measuring video quality or image quality is a complex process, even with the aid of human factor, since quality has not been defined effectively for dynamic environments. Approaches for quality evaluation are using the objective quality, which requires a number of computing algorithms, or the subjective quality, which requires manual effort [17]. In some cases a user or the system may specify quality threshold under which it is declared that no acceptable experience can be provided. However providers shall always deliver the best possible experience [8]. A user may use the mobile interactive browser for evaluating the adapted content by updating the *AdaptationQoS* parameters (indirectly). Quality of service is defined under MPEG-21 as *AdaptationQoS*. It describes the relationship between constraints, possible adaptations, and qualities in order to support media resource adaptation for terminal and network quality of service [4]. The *AdaptationQoS* descriptor provides the means to trade-off these parameters with respect to quality so that an adaptation strategy can be formulated and optimal adaptation decisions can be made in constrained environments. For example, a user can define the relative order of each conversion of an original modality and the numeric weight of each conversion (figure 3b). The weights of conversions with the *AdaptationQoS* parameters will help the selection process to determine when conversion should be made. *The Mobile digital item viewer* presents adapted content to users. Currently, the viewer supports audio, video, image and text delivery over http, https, rtp, rstp. For instance, a user can watch a movie as streaming video via the rstp protocol or can download it and watch it via the http protocol (figure 3c).

While a part of MPEG-21 framework is about user, terminal, network, usage, natural environment characteristics, it is unclear how these fit to a practical scenario or who and how is providing that information. These characteristics require specific treatment by specific actors, such as a mobile device, a user, or middleware. A user has to insert most of their data manually but the implementation of intelligent algorithms on a device or a middleware will automate this. The browser is able to retrieve device characteristics (e.g. device type, operating system display resolution, codec and content types supported by devices, processor, memory, etc) and use this information during digital item adaptation. Network characteristics can also be defined by the user manually or automatically by the device and middleware. The collaboration model, which illustrates the sources of “MPEG-21 characteristics”, is illustrated in figure 4a.

Figure 4b shows priority preference for general resources, where a user wishes to have high video QoS by assigning a *priorityLevel* of 1.5 to video resources. The user is also interested in Sports and gives a *priorityLevel* of 1.6 to this genre. So, the resources of video modality and Sports genre, especially the Sport videos, should have better qualities after adaptation. Note that the user already knows the default *priorityLevel* of resources is 1.0. However, the same user would specify different

weight values under different circumstances e.g. when he is in his/her car and he wants the content in less elapsed time. The priorityLevel is defined within user characteristics and is filled manually by a user (figure 4a, figure 3b).

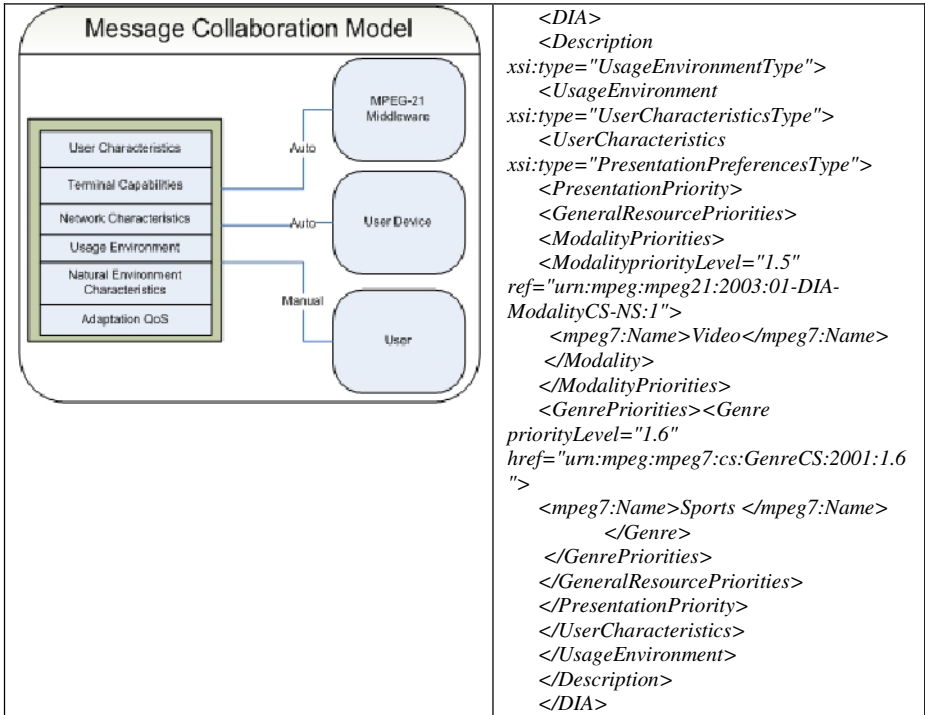


Fig. 4. a) Automatic and manual gathering of characteristics. b) MPEG-21 Presentation Priority.

### 4 Performance Evaluation and Complexity

During the performance evaluation of the architecture, a series of tests were performed on sample of user requests. A number of mobile phones having the multimedia mobile client were generating requests. Benchmarks revealed that the middleware performance is depended on the selection process and the selected adaptation operator. The selection process is heuristic, which makes it very difficult to determine and characterize precisely the execution time. Whilst each adaptation operator performance is different, it can be calculated. The main factors that affect each adaptation operator performance are the size of the digital item (e.g. video size), the initial and target format type and characteristics. The server’s configuration plays an important role for an adaptation operator (e.g. a dual 2,4 GHz CPU with two simultaneously real time transcoding channels may assist the transcoding operator to exceed real time). Most of the adaptation operation results can be predicted. For instance, consider the case of a mobile client which requests a video movie but the mobile device does not support the video format. Therefore, the

selection process finds an adaptation operation that can export to audio with parameters such as 8-bit/16-bit/32-bit sample type b. stereo/mono. These parameters may play an important role to the size of the adapted audio (figure 5). The final size can be calculated before adaptation takes place. However, some characteristics of the final digital item cannot be determined, such as the measurement of video, image or audio quality.

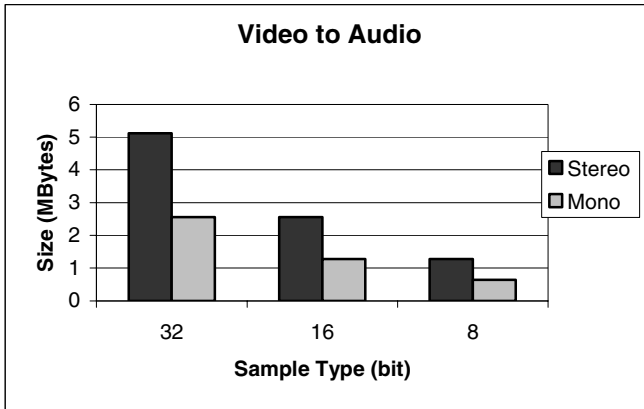


Fig. 5. Video (MPEG-2) to Audio adaptation

## 5 Conclusions and Future Work

In this paper we exploit how mobile computing can use MPEG-21 by illustrating a design of MPEG-21 middleware and a prototype multimedia mobile client. MPEG-21 mobile client allows a user to define his/her preferences and at the same time include mechanisms for retrieving and sending device and network information to the middleware. The middleware is designed to provide content satisfaction for most user preferences and device and network constrains. We are currently implementing the MPEG-21 middleware and its algorithms (adaptation, selector and evaluation algorithms).

## References

1. ISO/IEC, MPEG-21 Digital Item Adaptation, N5845, 2003
2. Burnett, I., Walle, R. V., Hill, K., Bormans, J.,Pereira, F.: MPEG-21: Goals and Achievements. *IEEE Multimedia*, Vol.10(4). (2003) 60-70
3. Chang, S. F.,Vetro, A.: Video Adaptation: Concepts, Technologies, and Open Issues. *Proceedings of the IEEE*, Vol.93(1). (2005) 148-158
4. Feiten, B., Wolf, I., Oh, E., Seo, J.,Kim, H.-K.: Audio Adaptation According to Usage Environment and Perceptual Quality Metrics. *IEEE TRANSACTIONS ON MULTIMEDIA*, Vol.7(3). (2005) 446-453
5. Ramesh, J.: A True Multimedia Client. *Multimedia IEEE*, Vol.12(2). (2005) 103-104

6. Tseng, L., Ching-Yung, L., Smith, J., R.: Using MPEG-7 and MPEG-21 for Personalizing Video. *IEEE Multimedia*, Vol.11(1). (2004) 42-52
7. Kim, J.-G., Wang, Y., Chang, S.-F.: Content-Adaptive Utility-Based Video Adaptation. *Proceedings of IEEE Int'l Conference on Multimedia & Expo*, Vol.3 (2003) 281-284
8. Pereira, F., Burnett, I.: Universal Multimedia Experiences for Tomorrow. *IEEE Signal Processing Magazine*, Vol.20(2). (2003) 63-73
9. Mukherjee, D., Delfosse, E., Kim, J.-G., Wang, Y.: Optimal Adaptation Decision-Taking for Terminal and Network Quality-of-Service. *IEEE TRANSACTIONS ON MULTIMEDIA*, Vol.7(3). (2005) 454-462
10. Timmerer, C., Hellwagner, H.: Interoperable Adaptive Multimedia Communication. *IEEE Multimedia*, Vol.12(1). (2005) 74-79
11. Panis, G., Hutter, A., Heuer, J., Hellwagner, H., Kosch, H., Timmerer, C., Devillers, S., Amielh, M.: Bitstream Syntax Description: A Tool for Multimedia Resource Adaptation within MPEG-21. *EURASIP Signal Processing: Image Communication Journal*, Vol.18(8). (2003) 721-747
12. De Keukelaere, F., De Zutter, S., Van de Walle, R.: MPEG-21 Digital Item Processing. *IEEE TRANSACTIONS ON MULTIMEDIA*, Vol.7(3). (2005) 427-434
13. Jerez, H., N., Liu, X., Hochstenbach, P., Van de Sompel, H.: The Multi-faceted Use of the OAI-PMH in the LANL Repository. *Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries*, (2004) 11 - 20
14. Kosch, H.: *Distributed Multimedia Database Technologies Supported by MPEG-7 and MPEG-21*. 1st edn. CRC PRESS, New York (2004)
15. Vetro, A., Timmerer, C.: Digital Item Adaptation: Overview of Standardization and Research Activities. *IEEE TRANSACTIONS ON MULTIMEDIA*, Vol.7(3). (2005) 418-416
16. Vetro, A.: MPEG-21 Digital Item Adaptation: Enabling Universal Multimedia Access. *IEEE Multimedia*, Vol.11(1). (2004) 84-87
17. Richardson, I. E. G.: *H.264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia*. 1st edn. John Wiley, Chichester (2003)

# A Unified Context Model: Bringing Probabilistic Models to Context Ontology\*

Binh An Truong, YoungKoo Lee\*\*, and Sung Young Lee\*\*

Department of Computer Engineering, KyungHee University,  
Giheung-Eup, Yongin-Si, Gyeonggi-Do, 449-701, Korea  
tabinh@oslab.khu.ac.kr, {yklee, sylee}@khu.ac.kr

**Abstract.** Ontology is a promising tool to model and reason about context information in pervasive computing environment. However, ontology does not support representation and reasoning about uncertainty. Besides, the underlying rule-based reasoning mechanism of current context-aware systems obviously can not reason about ambiguity and vagueness in context information. In this paper, we present an ongoing research on context modeling which follows the ontology-based approach while supports representation and reasoning about uncertain context. This unified context model then is used as a framework in our implementation of the context management and reasoning module of our context-aware middleware for ubiquitous systems.

## 1 Introduction

Most of the current proposed pervasive context-aware systems use sensors as the major source for providing data to applications. However, the data sensed, or raw data, is always imperfect and incomplete due to the sensing technologies. This result in the inaccuracy of the high-level information deduced from raw data. For example, it is difficult to infer that the user is sleeping based on the sensing data such as his location (in-bed), the room light (dark) and the sound (quiet). Furthermore the underlying logical, rule-based reasoning mechanism of current systems obviously does not support reason about uncertainty. Hence, dealing with uncertainty is the most challenge in context-aware computing research community.

Since its appearance, probabilistic model or Bayesian networks technique has showed to be a very powerful tool for the representation and reasoning about the uncertainty. In particular, a Bayesian network represents a full joint distribution over a set of random variables. It can answer queries about any of its variables given any evidence. Besides, Bayesian network provides different forms of reasoning including: prediction (reasoning from cause to result), abduction (inferring cause from result) and finally, explaining away (the evidence of one cause reduces the possibility of another cause given the evidence of their common results) which is especially difficult to model in rule-based systems [4]. Nevertheless, a fundamental limitation of

---

\* This work is partially supported by Korea Science and Engineering Foundation (KOSEF).

\*\* Corresponding authors.

using Bayesian network for knowledge representation is that it can not represent the structural and relational information. Also, the applicability of a Bayesian network is largely limited to the situation which is encoded, in advance, using a set of fixed variables. Thus, it is not suitable for representation of contextual data which is highly interrelated and dynamic in pervasive computing environment [5].

In this paper, we present a unified context model which inherits the advantages from both the probabilistic model and ontology. It can be considered as the glue which connects and integrates these two techniques. Given the unified context model, we can build a unified context ontology which captures both structural and probabilistic knowledge of a domain. Given the unified context ontology, Bayesian networks are constructed autonomously and used for reasoning about uncertainty.

The rest of paper is organized as follows. Section 2 discusses on the related work. Section 3 presents a scenario which will be used through this paper for illustrating of our approach. In section 4, we present our unified context model in detail. The unified context-ontology paradigm is introduced in section 5. Section 6 introduces three types of reasoning supported given the context ontology defined based on our context model. Finally, the paper ends with discussions and conclusions in section 7.

## 2 Related Work

A lot of research has addressed the issue of uncertainty in context aware computing. First efforts tried to modeling the uncertainty of context information using various terms such as "imperfectness" [8], "confidence" [9], "accuracy" [10], etc. Nevertheless, those approaches lack of expressiveness to capture rich types of context information and they do not support the reasoning mechanism.

Recently, some research approaches have used Bayesian networks to model and reason about the uncertain context. Firstly, Ranganathan et al. [2] used Microsoft's Belief Network (MSBN) [12] software to create the Bayesian networks structure. The Bayesian network is defined by knowledge experts and is mapped to predicates in the ontology by developers. Each predicate is attached with a confidence value for representing its value's uncertainty. Secondly, Tao Gu et. al. [11] mapped each context predicate into a node in the Bayesian network. Then, an arc is drawn between two nodes if and only if a dependency relation exists between two context predicates. Thus, a RDF graph with dependency markup is translated into a Bayesian network.

These two approaches have solved the problem of dealing with uncertainty. However, their support of uncertain reasoning is application-specific. In both approaches, developing a new application needs to redefine a new Bayesian network even if the domains are similar. Also, the mapping between the Bayesian network and the ontology is done manually by developers in both approaches. Even when the probabilistic data in form of Bayesian networks is integrated into the context ontology in [11], it is still unable to be reused for a similar domain. The reason is that it is defined over the instances data. In summary, both approaches provide no systematical method to support the uncertain reasoning mechanism.

### 3 A Smart-Home Scenario

In this section, we describe a smart home scenario which will be used for illustrating our proposed context model. Our sample scenario is a smart automated home that can proactively control the environmental conditions to reduce resource consumption. The windows and blinds can be controlled automatically according to the situations to provide optimal cooling or heating process or to create a fresh air breeze. For instance, on a day when the temperature is shifted from cool to warm, the home might determine that the optimal warming strategy is to open the windows and blinds so that the warm air can go inside. This scenario seems to be very simple but it is practically more complicated in the real situation. For example: the outside is so noisy while there are people reading inside the room; someone does not want the blinds open because he/she is sleeping; the air outside is polluted by dust and smoke, and so on. The decision of open the windows and blinds is depended on many elements in the situation. Such dependencies are also changed from situation to situation.

In the next section, we will show how to model this smart-home domain based upon our proposed unified context model

### 4 The Unified Context Model

Our context model is influenced mainly by the Probabilistic Relational Model developed by Friedman et al. in [8] and the Probabilistic Frame-based systems of Koller et al. in [9]. We made some modifications in compare with the original PRM to make the model simple and suitable to our requirements.

The unified context model consists of two parts:

- *The relational schema* which represents the structural and organizational information in forms of class, binary relations, relation chains and properties.
- *The probabilistic models* which annotate the conditional probabilistic dependency relationship between properties of classes.

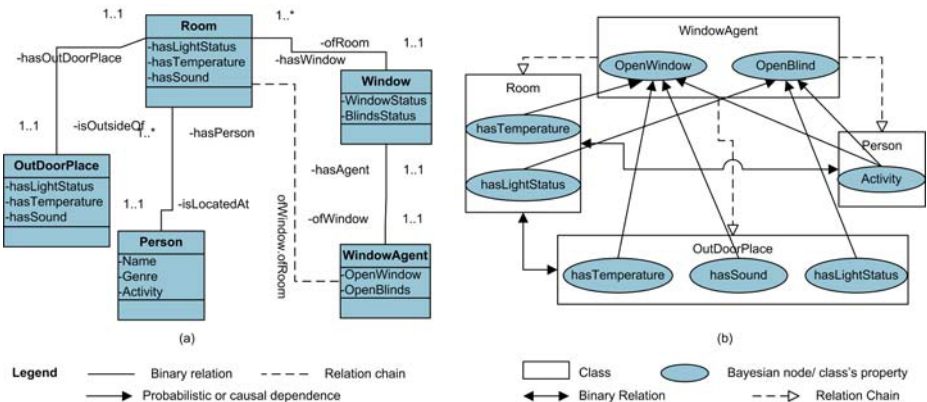


Fig. 1. (a) The relational schema and (b) the probabilistic model for the scenario



## 4.1 The Relational Schema

The basic unit of our context model is a class  $X$ . A class may be a sub-class of another (its super class). A class includes a set of relations  $R_1, \dots, R_n$  and a set of properties  $P_1, \dots, P_n$  with associated restrictions (or facet). A relation  $R_i$  specifies a binary relation between two classes  $X$  and  $Y$ . All relations are typed appropriately. The binary relation  $X.R(Y)$  can be considered as an object-property of class  $X$  which has the value-type of class  $Y$ . In Figure 1, the binary relation `hasWindow` defines the relationship between the class `Room` and the class `Window`.

A relation-chain is a sequence of binary relations separated by period. It creates an implicit relational link between two classes in a relational domain. A relation-chain  $X.R_1.R_2\dots R_n$  refers to the final class which is the type of the final relation in the chain. Each relation  $R_i$  in the chain must be correctly typed. In Figure 1, the chain `ofWindow.ofRoom` creates an indirect relation between class `WindowAgent` and class `Room`. Similarly, a property-chain is formed by appending a relation-chain with a property of the referenced class. It specifies a reference from a class to a property, which can be its property or other class property.

## 4.2 A Modified Probabilistic Relational Model

We use probability for representing the uncertainty within a domain. A class which consists of probabilistic information is annotated with the local probabilistic model. This type of class is called p-class. A p-class, similarly to the normal class, has properties, relations and restrictions.

We call the property which contains probabilistic information a p-property. A p-property is either simple or complex. A p-class may also have other properties that do not participate in the probabilistic model, whose type is neither of the above. This feature allows existing knowledge bases to be annotated with probabilistic information without requiring a complete redesign of the ontology.

A simple p-property corresponds with a root node in Bayesian network. A simple p-property has two restrictions: `hasValue` and `hasPD`. The restriction `hasValue` is an explicitly enumerated list of possible values for the p-property. The restriction `hasPD` specifies the probability distribution over the values listed in the `hasValue` restriction. For example, the p-property `hasTemperature` of the class `Room` may have the restriction `hasValue` as `{Hot, Warm, Cool, Cold}` and the restriction `hasPD` as `{0.3, 0.25, 0.25, 0.3}`. The sum of all probability values listed in a restriction `hasPD` must be equal to 1 to satisfy the probability axioms.

A complex p-property corresponds with a Bayesian network's node which has a set of parent nodes. Beside the two restrictions, `hasValue` and `hasPD`, a complex p-property has two other restrictions, `hasParents` and `hasCPT`, which specify the conditional probabilistic dependencies on other p-properties. The `hasParents` restriction of the complex p-property  $P$  specifies a list of property-chains on which the value of this property depends. Each property-chain refers to one property of other class. For example, in the `WindowAgent` class, the parent of the p-property `OpenWindow` may be the property-chain `ofWindow.ofRoom.hasTemperature`. The `hasCPT` restriction specifies the conditional probability distribution over the values of

the property given values of its parents, which are listed by the `hasParents` restriction. The conditional probability distribution is represented by using a conditional probability table (CPT) as in Bayesian networks. For each combination of values of its parents, the CPT provides a probability distribution over values of the property given its parents. For simplicity, we assume that the CPTs are represented as fully specified functions of parent values.

## 5 The Unified Context Ontology

Based on the proposed unified context model, we build a unified context ontology to capture the knowledge of the smart home domain as described in section 2. The `p`-class can be used just like any other normal class. We can create instances of class, which inherit all of its template properties and restrictions. In particular, the probability distribution over values of properties of the instance will be described in the `p`-class. Similarly, the inheritance mechanism can be used to make one `p`-class a subclass of another. A subclass can extend the definition of the super-class as well as overwrites parts of it. It can redefine the probabilistic model of one or more `p`-property.

The unified context ontology should be able to captures all the characteristics of context information. We classify the pervasive computing domain into a collection of sub-domains such as smart-home domain, smart-office domain, smart-university domain, etc. It would be easy to specify the context in one domain in which a specific range of context is of interest. The separation of domains can also reduce the burden of context processing.

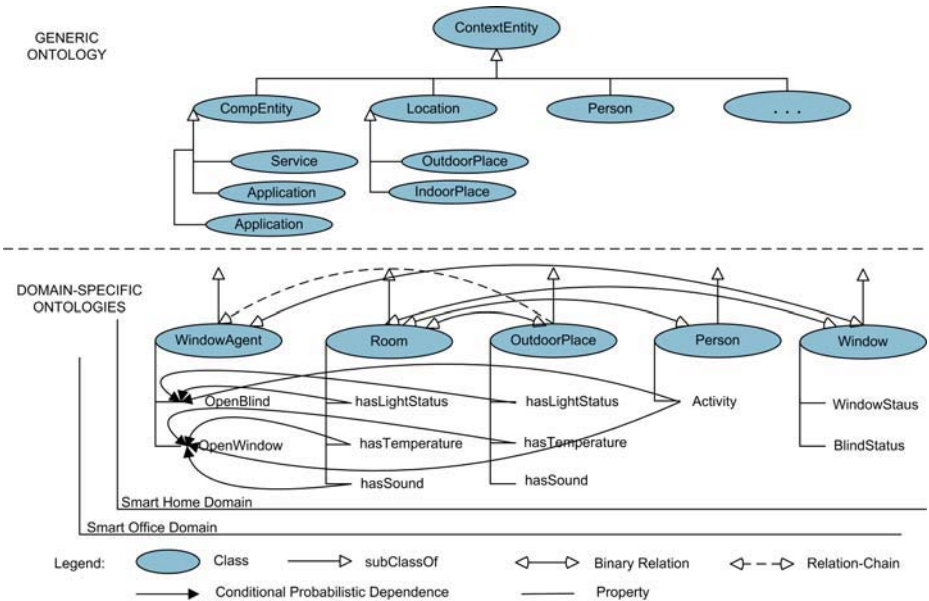


Fig. 2. A two-layer context ontology for relational and probabilistic knowledge

Our unified context ontology is divided into two layers including a generic ontology layer and a domain-specific ontologies layer as follows:

- *The generic ontology* is a high level ontology which captures general context knowledge about physical world in pervasive computing environment.
- *The domain-specific ontologies* are a collection of low-level ontologies which defines the details of concepts and properties in each sub-domain. A low-level ontology of a sub-domain consists of two parts: (1) relational schema which specifies relations and relation-chains of the sub-domain; and (2) probabilistic models which represent conditional probabilistic dependencies between properties in that sub-domain.

In Figure 2, the generic ontology defines basic concepts of CompEntity, Location, Person, Activity, etc. The details of each generic concept, such as relations, relation-chains, conditional probabilistic dependences, are redefined in domain-specific ontologies which may vary from one domain to another.

```

<owl:Restriction>
  <owl:onProperty>
    <owl:ObjectProperty rdf:resource="#OpenBlind" />
  </owl:onProperty>
  <rdf:hasParents>
    <rdf:List>
      <rdf:first rdf:resources="#PC-ofWindow.ofRoom.hasLightStatus"/>
      <rdf:rest> <rdf:List>
        <rdf:first rdf:resources="#PC-ofWindow.ofRoom.-
          hasOutdoorPlace.hasLightStatus" />
        <rdf:rest rdf:resource="&rdf:nil" />
      </rdf:List> </rdf:rest>
    </rdf:List>
  </rdf:hasParents>
  <rdf:hasCPT>
    <rdf:List>
      <rdf:first rdf:datatype="&xsd;integer">0.3</rdf:first><rdf:rest>
        <rdf:List>
          <rdf:first rdf:datatype="&xsd;integer">0.9</rdf:first><rdf:rest>
            <rdf:List>
              <rdf:first rdf:datatype="&xsd;integer">0.6</rdf:first>
              <rdf:rest>
                <rdf:List>
                  <rdf:first rdf:datatype="&xsd;integer">0.4</rdf:first>
                  <rdf:rest rdf:resource="&rdf:nil" />
                </rdf:List>
              </rdf:rest>
            </rdf:List>
          </rdf:rest>
        </rdf:List>
      </rdf:List> </rdf:rest>
    </rdf:List>
  </rdf:hasCPT>
</owl:Restriction>

```

**Fig. 3.** Example of the OWL-based probabilistic dependency relation

We also use the Web Ontology Language (OWL) [13] for representing context. However, we augmented new language elements to model new concepts such as relation-chain, property-chain and probabilistic dependency. We called this language

PROWL (Probabilistic annotated OWL). Figure 3 is an example of the PROWL-based ontology in which we use new markup elements .

## 6 Reasoning About Context

Based on the context ontology supports representation of both ontological and probabilistic knowledge, we could construct a knowledge base for a new application-domain. We support three reasoning types: rule-based reasoning, ontological reasoning and Bayesian reasoning.

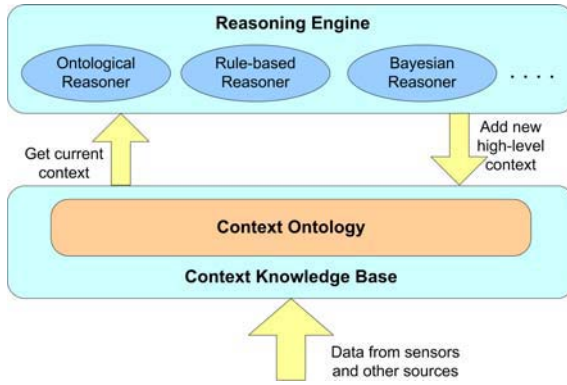


Fig. 4. Three supported reasoning mechanisms

### 6.1 Rule-Based and Ontological Reasoner

The rule-based reasoner is support by default given context ontology. However, there are differences when applying rules to a property of a p-class. The rule should update both the value of the property and the probability distribution over all values to satisfy the probability axioms.

For example, in our scenario, the context `WindowAgent.OpenBlind` can be deduced from the sensed, primary context `{OutdoorPlace.hasLightStatus, Room.hasLightStatus}` as follows:

$$\begin{aligned}
 & Prob(hasLightStatus(OutdoorPlace, Bright), 1.0) \wedge \\
 & Prob(hasLightStatus(Room, Dim), 1.0) \wedge \\
 & Prob(Activity(Binh, Sleeping), 1.0) \\
 \Rightarrow & Prob(OpenBlind(WindowAgent, Open), 1.0)
 \end{aligned}$$

The ontological reasoner can be described as an instance of the rule-based reasoner. However, it has a rule-base which consists of predefined rules to implement the language PROWL. In particular, the ontological reasoner can reason about OWL vocabularies and new concepts like relation-chain, property-chain. For example, if the class `WindowAgent` has a relation chain `ofWindow.ofRoom` which has type of the

class `Room`, the relation chain `ofWindow.ofRoom` of the instance `A` of class `WindowAgent` will refer to an instance `B` of class `Room` so that the relation-chain `ofWindow.ofRoom` satisfies.

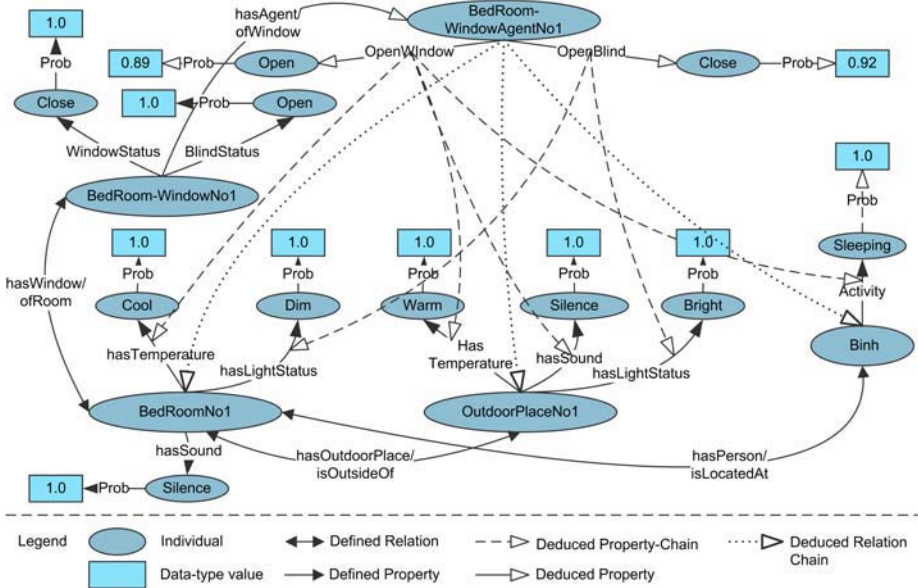


Fig. 5. An example of the context ontology for the scenario

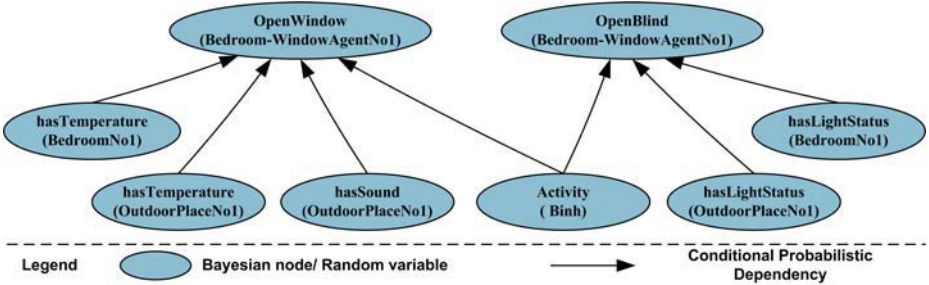


Fig. 6. A derived Bayesian network given the unified context ontology

### 6.2 Bayesian Reasoning

Before the standard Bayesian network inference can be used to answer queries about values of properties of instances, a Bayesian network is constructed from the context ontology. Depending on the domain, there may be more than one derived Bayesian network corresponding to each probabilistic relational model in the context ontology.

The algorithm **Construct-BN** [8] for deriving a Bayesian network is described as follows. Each node in the Bayesian net  $\mathcal{B}$  has the form  $\mathcal{I} . \mathcal{P}$  where  $\mathcal{I}$  is an instance of a  $p$ -class and  $\mathcal{P}$  is a property. The algorithm maintains a list  $\mathcal{L}$  of nodes to be processed. Initially,  $\mathcal{L}$  contains only the simple properties of named instances. In each iteration, the algorithm removes a node from  $\mathcal{L}$  and processes it. The removed node  $\mathcal{I} . \mathcal{P}$  is processed as follows. For each parent  $\mathcal{I} . \text{RC} . \mathcal{P}_i$ , which refers to a property of another instance, an edge is added from  $\mathcal{I} . \text{RC} . \mathcal{P}_i$  to  $\mathcal{I} . \mathcal{P}$ ; If  $\mathcal{I} . \text{RC} . \mathcal{P}_i$  is not already in  $\mathcal{B}$ , we add  $\mathcal{I} . \text{RC} . \mathcal{P}_i$  into  $\mathcal{B}$  and  $\mathcal{L}$ ; when all parents of  $\mathcal{I} . \mathcal{P}$  have been added, the CPT is constructed from the has-CPT restriction of  $\mathcal{I} . \mathcal{P}$ .

Since the Bayesian network is available, the standard Bayesian reasoner can use that network to infer about the probabilities of all nodes. Then, the probability of each node is updated directly to the property of instances the ontology.

We implemented the Bayesian reasoner based on the API of Microsoft Belief Network software [12]. The ontological and rule-based reasoners are developed based on the JenaAPI [14]. The mapping module for deriving Bayesian networks from the context ontology (implementing the Construct-BN algorithm) and updating new probability values to the ontology is also implemented based on Jena.

## 7 Discussions and Conclusions

The major characteristic in our approach is that we define the probabilistic information at the level of concepts. We not only specify the uncertainty of concept's value (property's value) but also specify the probabilistic or uncertain relationships between concepts. Since ontology mainly deals with concepts within a domain, our context model can easily extend the current ontology-based modeling approach. Based on our unified context model, we can easily define a unified, domain-oriented context ontology which captures both logical or relational and probabilistic knowledge. Given that unified context ontology, we can build several knowledge bases for similar applications. For example, we can model a smart-home domain and build the smart-home ontology. For every new smart-home applications, we only need to specify the instances given that predefined smart-home ontology without redefine or construct a new one. Besides, we can add probabilistic information into an existing ontology by adding relations, relation chains and restrictions without construct a new one from the scratch. Thus, our work in context modeling supports scalability and knowledge reusability. Since the mapping-relations between nodes in Bayesian networks and properties of classes are implicitly defined in the ontology, the mapping process can be programmed to run automatically. This feature reduces much burden on knowledge experts and developers in comparison with previous works [2], [11]. Finally, since probabilistic reasoning is supported, we can easily extend from reasoning to learning about uncertain context, which is simply learning about the parameters of Bayesian networks. The learning makes the Bayesian reasoning more robust and adaptive in highly dynamic and variable environments.

This paper describes our approach of representing and reasoning about uncertain context. Our study in this paper shows that the proposed context model is feasible and necessary for supporting context modeling and reasoning in pervasive computing. Our work is part of an ongoing research on Context Aware Middleware for Ubiquitous

System (CAMUS), which attempts to provide an easy, reusable infrastructure to develop ubiquitous context-aware applications. We are exploring methods to integrate multiple reasoning methods from AI area and their supported representation mechanism into the context reasoning and management layer.

## References

- [1] Satyanarayanan, M, "Coping with uncertainty", IEEE Pervasive Computing, page 2, Volume 2, Issue 3, July-Sept. 2003
- [2] Anand Ranganathan, Jalal Al-Muhtadi, Roy H. Campbell, "Reasoning about Uncertain Contexts in Pervasive Computing Environments", IEEE Pervasive Computing, pp 62-70 (Vol.3, No 2) , Apr-June 2004.
- [3] Abdelsalam, W.; Ebrahim, Y., " Managing uncertainty: modeling users in location-tracking applications", IEEE Pervasive Computing, pages 60-65, Volume 3, Issue 3, July-Sept. 2004
- [4] J. Pearl, "Belief Networks Revisited". In Artificial intelligence in perspective, pages 49-56, 1994
- [5] Henriksen, Karen and Indulska, Jadwiga and Rakotonirainy, Andry., "Modeling Context Information in Pervasive Computing Systems". First International Conference on Pervasive Computing, Pervasive'2002, LNCS(2414), pages 167-180, Zurich, August 2002.
- [6] Nir Friedman , Lise Getoor , Daphne Koller and Avi Pfeffer, "Learning Probabilistic Relational Models", Proceedings of the 16th International Joint Conference on Artificial Intelligence (pp. 1300-1307), Stockholm, Sweden, August 1999.
- [7] Daphne Koller and Avi Pfeffer, "Probabilistic frame-based systems", Proceeding of the 15th National Conference on Artificial Intelligence (pp. 580-587), Madison, Wisconsin, July 1998.
- [8] Gregory D. Abowd and Anind K. Dey, "Towards a Better Understanding of Context and Context-Awareness", Workshop on the what, who, where, when and how of context-awareness at CHI 2000, April 2000.
- [9] Hui Lei, Daby M. Sow, John S. Davis, II, Guruduth Banavar and Maria R. Ebling, "The design and applications of a context service", ACM SIGMOBILE Mobile Computing and Communications Review, vol 6, no. 4, pp 44-55, 2002.
- [10] Gray, P., Salber, D. "Modeling and using sensed context in the design of interactive applications", In Proceedings of 8th IFIP Conference on Engineering for Human-Computer Interaction, Toronto, 2001.
- [11] Tao Gu1, Hung Keng Pung and Da Qing Zhang, "A Bayesian approach for dealing with uncertain contexts", Proceedings of the Second International Conference on Pervasive Computing (Pervasive 2004), Vienna, Austria, April 2004.
- [12] Microsoft Belief Network software, <http://research.microsoft.com/adapt/MSBNx/>
- [13] W3C, "Web Ontology Language (OWL)", <http://www.w3.org/2004/OWL/>
- [14] Jena, "A Semantic Web Framework for Java", <http://jena.sourceforge.net/>

# A Component-Based Adaptive Model for Context-Awareness in Ubiquitous Computing

Soo-Joong Ghim<sup>1</sup>, Yong-Ik Yoon<sup>1</sup>, and Ilkyeun Ra<sup>2</sup>

<sup>1</sup> Department of Computer Science, Sookmyung Women's University,  
Chungpa-Dong 2-Ga, Yongsan-Gu, 140-742, Seoul, Korea  
{yiyoon, sjghim}@sookmyung.ac.kr

<sup>2</sup> Department of Computer Science & Engineering,  
University of Colorado, Denver and Health Sciences Center,  
Denver, CO 80217, U.S.A.  
ikra@carbon.cudenver.edu

**Abstract.** A high adaptable middleware has been an essential platform to provide more flexible services for multimedia, mobile, and ubiquitous applications in ubiquitous computing environments. In addition, the persistent services of these application systems and their middleware in ubiquitous computing are required so that they can be aware of the frequent and unpredictable changes in users' requirements as well as environmental conditions and adapt their behavioural changes. However, current approaches for supporting adaptability have made applications themselves trigger and execute an adaptive mechanism when the underlying infrastructure notifies them about any changes. In this paper, we propose a novel component-based context-adaptive model for context-awareness middleware to support efficiently dynamic adaptation of application services. We also demonstrate the current implementation of the context-adaptive middleware that help applications to adapt their ubiquitous computing environments according to rapidly changing contexts such as user-specific preferences, application-specific preferences, and low-level configurations.

## 1 Introduction

Ubiquitous computing allows application developers to build a large and complex distributed system that can transform physical spaces into computationally active and intelligent environments [2]. Ubiquitous applications need a middleware that can detect and act upon any context changes created by the result of any interactions between users, applications, and surrounding computing environment for applications without users' interventions [1]. The context-awareness has become the one of core technologies for application services in ubiquitous computing environment and been considered as the indispensable function for ubiquitous computing applications. In order to provide context-awareness services, underlying platforms in ubiquitous computing should be able to recognize contextual changes so that applications use contexts for evaluating new environments and finding an appropriate action by the result of evaluation for these changes.



Previous context-awareness researches have been mainly focusing on the location change of users or devices in context-awareness related fields [1,6]. In ubiquitous computing environment, however, devices or software services can be added to or removed from the system at anytime, and also contexts or preferences of users are changing frequently. Thus, the mobile applications are expected to be capable of adapting their behaviors to ensure they can continue to offer the best possible level of service to their users. The required level of a system's adaptation can be ranged from the operating system up to the application level.

Supporting high adaptability is an essential requirement for ubiquitous computing systems because the ubiquitous computing environments is highly dynamic, and characterized by frequent and unpredictable changes in different contexts. More generally, the adaptation can be applied to a wide range of aspects at different levels of the system. These include the communications aspects as well as several issues such as allocating resources to the set of activities and external services currently being used. Thus, this adaptation should be driven by awareness of a wide range of issues including its communication performance, resource usage, location, cost, and preference [3]. The extreme emphasis on adaptation at specific level can cause some problems, for example, the adaptation at the operating system level can serious affect on its integrity and performance. In opposite case, leaving all adaptations to the application level would impose a heavy burden on application programmers.

The current approach for supporting adaptable services or applications is based on the classic layered architectural model where adaptation is provided at the various layers (data link, network, transport or application layers) in isolation [4]. Due to the very limited information being shared across protocol layers, adaptation strategies are *uni-dimensional* – i.e. they only consider one parameter at a time – and often are leading to unsatisfactory results [5]. Another limitation of current approaches is that applications themselves are responsible for triggering and executing adaptive mechanism when the underling infrastructure notifies them about any changes [6].

In this paper, we present a novel adaptive model for context-awareness middleware framework that can overcome the current limitations and support persistence services that allow mobile applications to operate well balanced adaptations level between system and application, and customize the adaptation services for contextual changes. We also describe a component-based context-adaptive model to manage dynamic adaptation in contextual changes. The rest of this paper is organized as follows: Section 2 proposes a component-based framework to support adaptability for mobile agents. In section 3 we explain our conceptual context-adaptive model and component model for ubiquitous computing. Section 4 shows how adaptations are triggered by a policy and explain the detail implementations for audio service for PDA users. Section 5 concludes our research results and discusses future works.

## 2 A Context-Adaptive Model

### 2.1 Definition and Classification of Context

A term called context has different meaning, and is used extensively. Meaning of a context has been used in many various research fields such as operating systems, programming language, artificial intelligences, and natural language processing.

Moreover, the meaning of context in a user interface design is very different from those areas. It, however, has been agreed on that a context is the key concept for mobile application systems in ubiquitous computing, although it is differently understood from and used by other various fields.

In this paper, we define contexts as environmental information that may cause to activate adaptation mechanisms and as requirements of users or applications. We classify contexts into two levels: high-level and low-level context in Figure 1. The high-level contexts include the preference of users and applications. The preference is the explicit requirements about resources and quality of services. The resources preference is a use degree of resources that are requested by users or applications. The quality of service preference includes display elements such as resolution, a frame rate, and users' security requirements. The low-level contexts are subdivided into the user configuration and resources configuration. The user configuration consists of users' device, location, time. The resource configuration is systems' memory size, computing power, and network bandwidth.

Current existing other research efforts have considered the user preference as a kind of the information that a user would like to get information or contents, and can be used to supply appropriate information to the user through filtering by similarity measurement. However, we take into consideration of characteristics of ubiquitous computing, and we define the user preference as 'the explicit requirements for contexts' in order to adapt dynamically delivery of services when context changes are occurred.

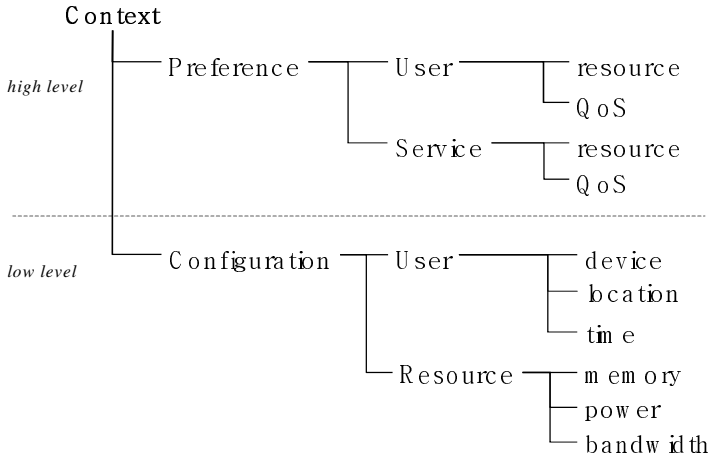


Fig. 1. Classification of context

## 2.2 Conceptual Model

### 2.2.1 Adaptation

Our context-adaptive model defines several elements for supporting context-awareness and adaptation methods. The detail relationships between elements of the context-adaptive model are shown in Figure 2.

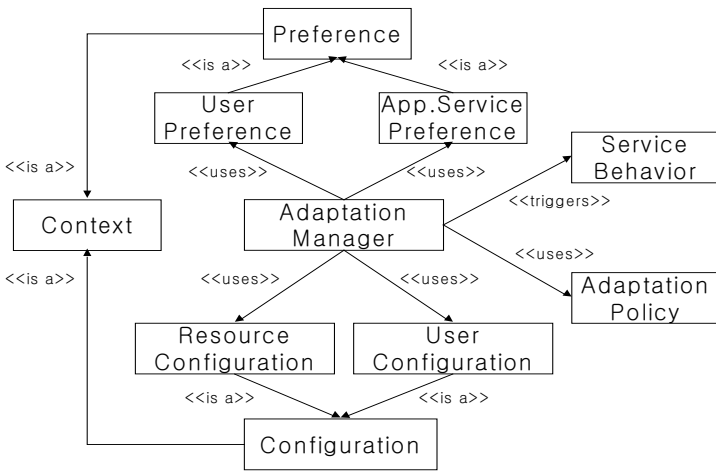


Fig. 2. Adaptation model

As shown in Figure 2, both the user preference and the application preference are considered as the explicit requirements for the usage of resources. We number the associative relationship between these preferences in order to decide on how to provide application services that can maximize satisfaction of user requirements. The configuration information includes a user’s device types, locations, time, resources, context of user devices, and adaptation policies that specify association rules between contextual changes of resources and application service behaviours. In our model, the adaptation method reflects preference and configurations, and satisfies user requirements that have user environments to be safe and operational.

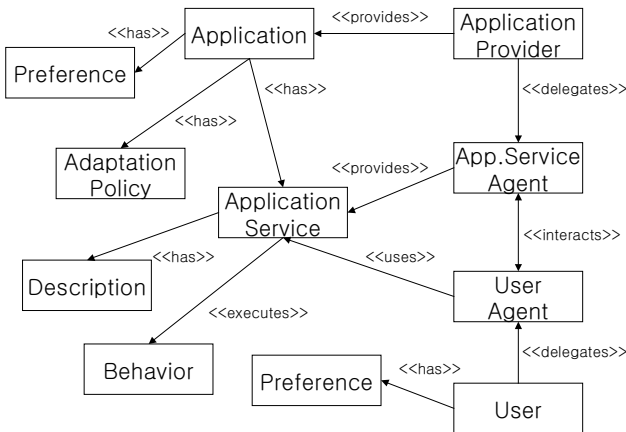


Fig. 3. Application service model

### 2.2.2 Application Service

Our application service model emphasizes on individual behaviours that compose application services, and establish details related to application service offering as shown in Figure 3.

An application service is the abstract concept of a task to be performed, and can be expressed with the description about a name and functions. The functionality of task is realized by service behaviours that can be composed of operations or methods that are invoked by an agent. In this model, an application consists of application services, adaptation polices and preference for users and services. The adaptation polices in an application are used for the context adaptation.

## 3 Dynamic Context-Adaptation

### 3.1 Adaptation Policy

The adaptation policy is a set of rules for changing application service behaviors with respect to different contexts, and is a key feature for the triggered adaptation. The triggered adaptation does not depend on only the notification mechanism from an underlying system as a voluntary adaptation way of an application about a context. It depends upon the self and dynamic detection for changes in an application behavior according to contextual changes or an application’s needs related to controlling an adaptation process.

To support context-awareness, it is required to process low-level tasks including periodic monitoring and detection of changes. A middleware can conceal the complexity with carrying out this task, and can decrease a burden of application developer. However, applications should inform a middleware about which a service should be adapted, and which a behavior should be triggered for each specific context when applications are being instantiated because it is difficult for a middleware to predict various changes of each context in requirements of an application.

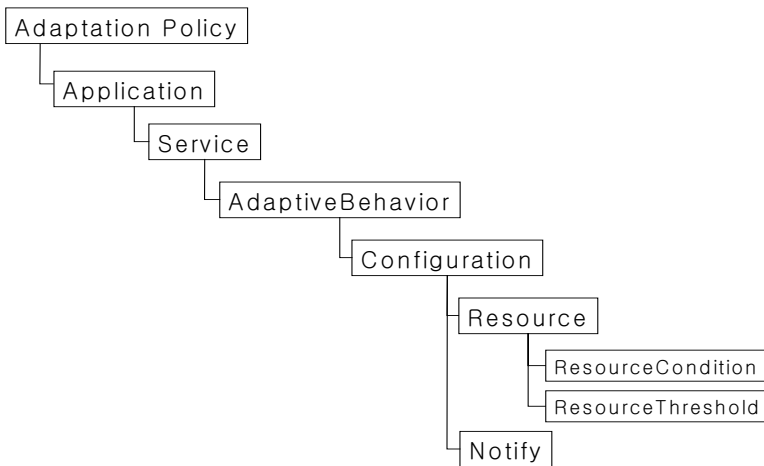


Fig. 4. Adaptation policy hierarchy

As shown in Figure 4, the adaptation policy describes important information for application services, required configuration in offering services, and behaviors to be triggered when certain context changes occur. A middleware would decide which rule is to apply for service delivery in current context using adaptation policy when application service is requested.

```

BehaviorSelection()
  Upriori ← ∅
  Spriori ← ∅
  Max ← 0
  Cost ← 0
  // compute priority of preference
  for all UserPreference
    Upriori ← Upriori ← Priority(UserPreference)
  for k ← 1 to numberOfBehavior
    for all AppServicePreference
      Spriori[k] ← Spriori[k] ← Priority(AppServicePreference)
  for k ← 1 to numberOfBehavior
    //compute the distance matrix Md,
    for all x ← Upriori and i ← 1 to numberOfPreferenceItem
      for y ← Spriori[k] and j ← 1 to numberOfPreferenceItem
        Md[i,j] ← (x - y)
    //compute the weight matrix Mw
    for i ← 1 to numberOfPreferenceItem
      for j ← i to numberOfPreferenceItem
        Mw[i,j] ← wij
        Mw[j,i] ← wji
    //compute matrix Mc
    for all p ← Upriori and i ← 1 to numberOfPreferenceItem
      for j ← 1 to numberOfPreferenceItem
        Mc[i] ← Mw[i,j] / ← Mw[i,j] ← p
    //compute cost function
    for i ← 1 to numberOfPreferenceItem
      Cost ← ← Mc[i]
    //select candidate behavior
    if(Max ← Cost) then
      Max ← Cost
    // Behavior with the maximum cost
    candidateBehavior ← Behavior

return candidateBehavior

```

**Fig. 5.** High level adaptation for the behavior selection

### 3.2 Adaptation Method

With triggered-adaptation, we support an application adaptation for context changes; context-changes trigger adaptation. The way of adaptation is carried out when an event is driven by applications. Applications can dynamically change the set of possible behaviors, as well as the associations between these behaviors and their corresponding enabling contexts, in order to cope with varying application needs and unpredicted context conditions. For the adaptation, we propose behavior decision algorithm that is composed of behavior selection algorithm and behavior mediation algorithm.

#### 3.2.1 Behavior Selection

Generally mobile users would require higher QoS as possible with the minimal resource consumption, because mobile devices have the limited computing and network resources. At the selection stage, the adaptation manager in middleware measures the fitness of application to user requirements for regulating the tradeoff of these two objectives. The following describes our high-level adaptation: behavior selection (See Fig. 5). Firstly, the adaptation manager performs the ranking computation to get priority of preference, and then compute the distance matrix  $\leftarrow_d$ . Each element  $\leftarrow_d$  represents interval between user's preference and application's preference. Based on  $\leftarrow_d$ , the weight matrix  $M_w$  is calculated, which represents how much  $A$  is preferred than  $B$ . In the current work, we use experimental values for the preferred degree,  $M_w$ . The cost matrix is  $M_c$  computed from  $M_w$ , to valuate the fitness of application service behavior. Hence, the behavior with the maximum cost is selected as the candidate behavior.

```

BehaviorMediation(candidateBehavior)
  minCost ← ←
  Solution ← ∅
  if all CurrentConfig ← upperBound(candidateBehavior)
    and CurrentConfig ← lowerBound(candidateBehavior) then
    adaptiveBehavior ← candidateBehavior
  else
    //find Behaviors satisfy CurrentConfig
    for all Behavior except CandidateBehavior
      if all CurrentConfig ← upperBound(Behavior)
        and CurrentConfig ← lowerBound(Behavior) then
        Solution ← Solution ← Behavior
    for all Behavior ← Solution
      for all Config ← Policy(Behavior)
        mediationCost ← median([upperBound-lowerBound]
          -CurrentConfig)
        if minCost ← mediationCost then
          minCost ← mediationCost
          adaptiveBehavior ← Behavior
  return adaptiveBehavior
  
```

Fig. 6. Low-level adaptation for the behavior mediation

### 3.2.2 Behavior Mediation

To execute the candidate behavior selected at the previous stage, the current configuration should support this behavior. On this mediation phase, we examine the current configuration of user's device and then decide whether the candidate behavior can be executed or not. The following describes low-level adaptation: behavior mediation (See Fig. 6).

If the candidate behavior is not suitable for current conditions, the adaptation manager should find another behavior to adapt. In the other words, if the middleware has more than one behavior whose upper threshold and lower threshold satisfying current condition, the middleware compute **mediationCost** to select the behavior with the minimum cost as the adaptive behavior.

## 4 Implementation

We now present our current implementation of the context-adaptive middleware. We have implemented middleware engine and metadata server in Java using jdk 1.4.2 and MySQL 4.0. At this stage, we have demonstrated audio service for PDA users.

Figure 7 represents an overview of our system architecture. Configuration Manager receives configuration information from sensor nodes whenever user configurations are changed. Adaptation Manager performs the behaviour decision process using context information; preferences and configurations. Execution Manager maintains binding information of behaviors to execute services. When an adaptive behavior is selected in the result of the behavior decision, Execution Manager binds behavior objects to provide mobile users with their application services. Communication Manager is responsible for exchanging XML messages to store or retrieve metadata.

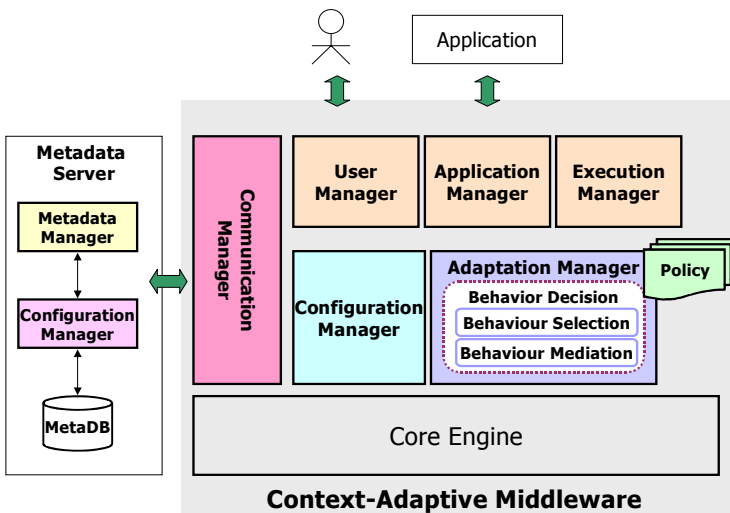
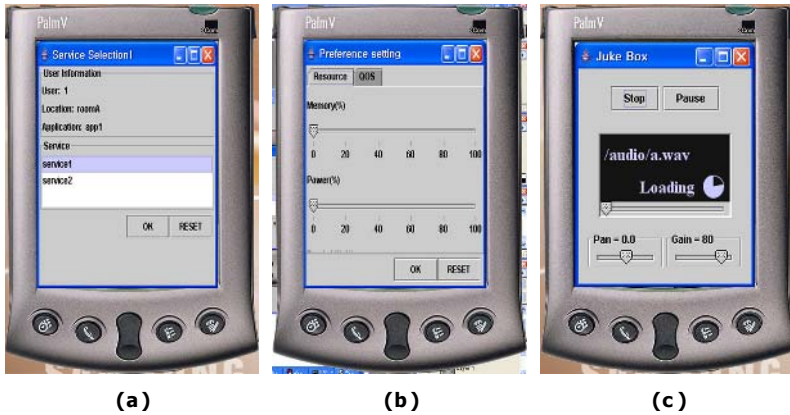


Fig. 7. Context-Adaptive Middleware Architecture



**Fig. 8.** View of Implementation Results. (a) Service Selection, (b) Preference Setting, (c) Service screen.

In Figure 8, there are shown three kinds of implementation results. Figure 8(a) shows the interface of service selection and Figure 8(b) shows the interface of preference setting for user requirement. The screen in Figure 8(c) shows the loading of contents to support an application service by managing of the context-awareness middleware.

## 5 Conclusions and Future Work

The context awareness is the essential technology that middleware must be equipped with in the ubiquitous environment where various computing unit are connected through a various types of network in order to provide continuous service to a user. Therefore, the middleware for applications in the ubiquitous computing environments should easily acquire context information from various context sources and should provide service adapted to context changes.

In this paper, we defined a context itself and described the design of context-adaptive model based on component for applications in ubiquitous environments. The context adaptive model can support dynamic adaptation for mobile users and applications using the context such as high-level and low-level information. We proposed a policy-based adaptation method using mobile agents in context-awareness middleware. The adaptation policies can be specified by the user-specific and application-specific priorities on a user’s preference to applications and quality of services, and an application’s resource requirements. The implementation of our adaptive middleware framework is currently ongoing, focusing on supporting context-aware mobile applications. In the future work, we intend to develop adaptive middleware services and management mechanism for context information. The adaptive model will be used for a contents rendering function that is a part of context-awareness middleware.

**Acknowledgement.** This research was supported by IRC(Internet Information Retrieval Research Center) in Hankuk Aviation University. IRC is a Kyonggi-Province Regional Research Center designated by Korea Science and Engineering Foundation and Ministry of Science & Technology.



## References

1. A. K. Dey, D. Salber, and G. D. Abowd, A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-Computer Interaction*, Vol. 16, 2001.
2. A. Ranganathan and R. H. Campbell, A Middleware for Context-Aware Agents in Ubiquitous Computing Environments, In *ACM/IFIP/USENIX International Middleware Conference*, Rio de Janeiro, Brazil, June 16-20, 2003.
3. G. S. Blair, G. Coulson, A. Anderson, et. al., A Principles Approach to Supporting Adaptation in Distributed Mobile Environments. *Proceedings of the 5th International Symposium on Software Engineering for Parallel and Distributed Systems (PDSE'2000)*, Nixon P. & Ritchie I. (eds), Limerick, Ireland, June 10-11, 2000.
4. Z. J. Haas, Designing Methodologies for Adaptive and Multimedia Networks, *IEEE Communications Magazine*, pp. 106-107, Vol. 39, N.11, November 2001.
5. A. Liotta, A. Yew, C. Bohoris, and G. Pavlou, Supporting Adaptation-aware Services through the Virtual Home Environment, *Proceedings of the 9th Workshop of the HP OpenView University Association*, June 11-13, 2002.
6. C. Efstratiou, K. Cheverst, N Davices and A. Friday, An Architecture for the Effective Support of Adaptive Context-Aware Applications, *Proceedings of the Second International Conference on Mobile Data Management (MDM '2001)* , pp. 15-26, January 8 - 10, 2001.
7. M. Román, F. Kon, and R. H. Campbell, Reflective Middleware: From Your Desk to Your Hand, *IEEE Distributed Systems Online Journal*, Special Issue on Reflective Middleware, Vol. 2 , No. 5, 2001.
8. B. Noble, System Support for Mobile, Adaptive Applications, *IEEE Personal Communications*, Vol. 7, No. 1, February 2000.
9. Capra, L., Blair, G. S., Mascolo, C., Emmerich, W., and Grace, P., Exploiting Reflection in Mobile Computing Middleware, *ACM Mobile Computing and Communications Review*, 6(4), pp. 34-44, 2003.

# Improvement of an Efficient User Identification Scheme Based on ID-Based Cryptosystem

Eun-Jun Yoon and Kee-Young Yoo\*

Department of Computer Engineering, Kyungpook National University,  
Daegu 702-701, Republic of Korea  
Tel.: +82-53-950-5553, Fax: +82-53-957-4846  
ejyoon@infosec.knu.ac.kr, yook@knu.ac.kr

**Abstract.** In 2004, Hwang et al. proposed an efficient user identification scheme based on an ID-based cryptosystem that is suitable for the wireless/mobile environment. However, we find that their scheme is still vulnerable to impersonation attack. Accordingly, the current paper first shows the weakness of Hwang et al.'s scheme, and then presents an enhancement to resolve such problems.

**Keyword:** Mobile network, Security, Identity-based cryptosystem, User identification, Impersonation attack.

## 1 Introduction

In 1984, Shamir first proposed a user-identification scheme based on a public key cryptosystem that can let each user's identification information be his public key [1]. In this system, the public key of each entity is an identity which can be defined as that part of the identification information. In 1991, Maurer-Yacobi [2] proposed an identity-based non-interactive public key distribution system based on a novel trapdoor one-way function of exponentiation modulo of a composite number. In 1998, Tseng-Jan [3] modified a non-interactive public key distribution system and proposed several applications based on the Maurer-Yacobi scheme.

Recently, Hwang et al. [4] proposed an efficient scheme based on Tseng-Jan scheme that is more suitable for the wireless environment. Compared with Tseng-Jan scheme, Hwang et al.'s scheme reduces the time required for responding and waiting. Additionally, their scheme uses the user's identity as his or her public key and does not need a key directory to store users' keys. However, we have found that their scheme is still vulnerable to impersonation attacks. Accordingly, the current paper first shows the weakness of Hwang et al.'s scheme, and then presents an enhancement to resolve such problems.

This paper is organized as follows: In Section 2, we briefly review Hwang et al.'s scheme. Section 3 shows how it suffer from vulnerability to impersonation attacks. In Section 4, we present an improvement of Hwang et al.'s scheme. In Section 5, we analyze the security of our proposed scheme. Finally, our conclusions are given in Section 6.

---

\* Corresponding author.

## 2 Review of Hwang et al.'s Scheme

Hwang et al.'s scheme is composed of three phases: system initialization, user registration, and user identification.

### 2.1 System Initialization Phase

A trusted authority generates system parameters as follows:

- $M$ : the mobile device.
- $ID_m$ : the mobile device's identity.
- $BS$ : the base station.
- $ID_b$ : the base station's identity.
- $N$ : the product of four primes  $p_j$ , for  $1 \leq j \leq 4$ , whose decimal digits are between 60 and 70 and the numbers  $(p_j - 1)/2$  are odd and pairwise relatively prime.
- $e$ : a public integer in  $Z_{\phi(N)}^*$ .
- $d$ : a secret integer, which satisfies  $ed \equiv 1 \pmod{\phi(N)}$ .
- $t$ : a random number from  $Z_{\phi(N)}^*$ , where  $\phi$  is the Euler's totient function.
- $s_m$ : a secret key of  $M$ , which satisfies  $s_m = e \cdot t \cdot \log_g(ID_m^2) \pmod N$ .
- $s_b$ : a secret key of  $BS$ , which satisfies  $s_m = e \cdot t \cdot \log_g(ID_b^2) \pmod N$ .
- $g$ : a primitive element in  $GF(p_j)$ .
- $h(\cdot)$ : a one-way function.
- $||$ : a bits connection.
- $T$ : a timestamp.

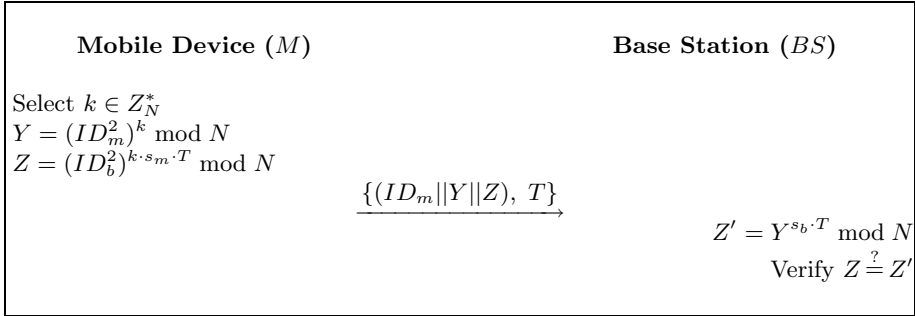
### 2.2 User Registration Phase

When mobile device ( $M$ ) joins the system, the procedures of user registration phase is shown as follows:

- Step 1.  $M$  presents his or her unique identity  $ID_m$  to the trusted authority.
- Step 2. The trusted authority computes  $s_m = e \cdot t \cdot \log_g(ID_m^2) \pmod N$ ,  $v = t^{-1} \pmod{\phi(N)}$  and sends  $s_m$  to  $M$  as his or her secret key. Finally, the trusted authority publishes  $\{N, g, e, h(\cdot)\}$  and keeps  $\{p_1, p_2, p_3, p_4, t, v, d\}$  secret for all users.
- Step 3.  $M$  publishes  $\{ID_m\}$  and keeps  $\{s_m\}$  secretly.

### 2.3 User Identification Phase

Hwang et al. proposed a user identification scheme only using one pass to show the validity of a user's identity. Figure 1 shows Hwang et al.'s user identification phase. When the mobile device ( $M$ ) wants to show his identity  $\{ID_m\}$  to the verifier the base station ( $BS$ ), the procedures of user identification phase is shown as follows:



**Fig. 1.** Hwang et al.'s User Identification Phase

Step 1.  $M$  chooses a random integer  $k$  in  $Z_N^*$  and computes  $Y$  and  $Z$  as follows:

$$Y = (ID_m^2)^k \bmod N,$$

$$Z = (ID_b^2)^{k \cdot s_m \cdot T} \bmod N.$$

Then,  $M$  sends  $\{(ID_m || Y || Z), T\}$  to  $BS$ .

Step 2. After receiving the above messages from  $M$ ,  $BS$  computes  $Z' = Y^{s_b \cdot T} \bmod N$ , where  $s_b (= e \cdot t \cdot \log_g(ID_b^2) \bmod N)$  is  $BS$ 's secret key.

Step 3.  $BS$  checks the equation  $Z \stackrel{?}{=} Z'$ . If the equation holds,  $BS$  will confirm that  $M$ 's identity is valid.

### 3 Impersonation Attack on Hwang et al.'s Scheme

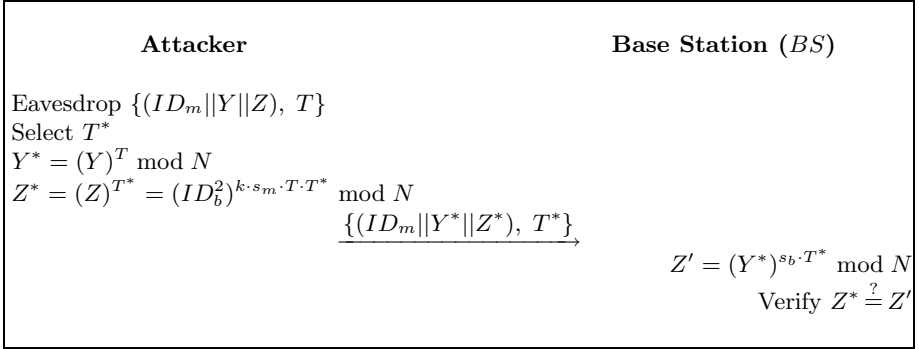
In Hwang et al.'s scheme, an attacker can easily impersonate a mobile device ( $M$ ). Suppose that the attacker has eavesdropped a valid message  $\{(ID_m || Y || Z), T\}$  from the network. It is easy to obtain the message since they are exposed over the open network. Then the impersonation attack proceeds as follows:

Step 1\*. The attacker select a timestamp  $T^*$  and computes  $Y^* = (Y)^T \bmod N$  and  $Z^* = (Z)^{T^*} = (ID_b^2)^{k \cdot s_m \cdot T \cdot T^*} \bmod N$ . Then, the attacker sends  $\{(ID_m || Y^* || Z^*), T^*\}$  to the base station ( $BS$ ).

Step 2\*. After receiving the attacker's message,  $BS$  will compute  $Z' = (Y^*)^{s_b \cdot T^*} \bmod N$  and check the equation  $Z^* \stackrel{?}{=} Z'$ .

We can easily check that  $BS$  will accept this message as follows:

$$\begin{aligned} Z^* &= (ID_b^2)^{k \cdot s_m \cdot T \cdot T^*} \bmod N \\ &= (g^{v \cdot s_m \cdot d})^{k \cdot s_b \cdot T \cdot T^*} \bmod N \\ &= (ID_m^2)^{k \cdot s_b \cdot T \cdot T^*} \bmod N \\ &= (Y^*)^{s_b \cdot T^*} \bmod N \\ &= Z' \end{aligned}$$



**Fig. 2.** Impersonation attack on Hwang et al.'s User Identification Scheme

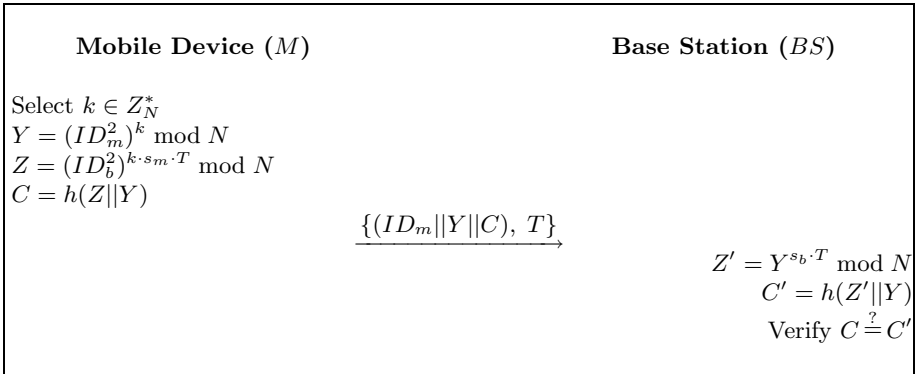
In is thus apparent that Hwang et al.'s user identification scheme is insecure. Figure 2 shows an impersonation attack on Hwang et al.'s user identification scheme.

## 4 Improvement of Hwang et al.'s Scheme

In order to solve the weakness mentioned in Section 3, we propose an improved scheme. System initialization and the user registration phase are equal to Hwang et al.'s scheme. Figure 3 shows the proposed user identification phase. The procedures of proposed user identification phase are as follows:

Step 1. Mobile device ( $M$ ) chooses a random integer  $k$  in  $Z_N^*$  and computes  $Y$ ,  $Z$ , and  $C$  as follows;

$$\begin{aligned}
 Y &= (ID_m^2)^k \bmod N, \\
 Z &= (ID_b^2)^{k \cdot s_m \cdot T} \bmod N, \\
 C &= h(Z || Y).
 \end{aligned}$$



**Fig. 3.** Proposed User Identification Phase

Then,  $M$  sends  $\{(ID_m||Y||C), T\}$  to the base station( $BS$ ).

Step 2. After receiving the above messages from  $M$ ,  $BS$  computes  $Z'$  and  $C'$  as follows;

$$\begin{aligned} Z' &= Y^{s_b \cdot T} \bmod N, \\ C' &= h(Z' || Y). \end{aligned}$$

Step 3.  $BS$  checks the equation  $C \stackrel{?}{=} C'$ . If the equation holds,  $BS$  will confirm that  $M$ 's identity is valid.

## 5 Security Analysis

In this section, we shall only discuss the enhanced security features. The rest are the same as original Hwang et al.'s scheme as described in literature [4].

**Definition 1.** A secure one-way hash function is a function  $f$  such that for each  $x$  in the domain of  $f$ , it is easy to compute  $f(x)$ ; but for essentially all  $y$  in the range of  $f$ , it is computationally infeasible to find any  $x$  such that  $y = f(x)$  [5].

**Theorem 1.** In the proposed user identification phase, an attacker cannot impersonate a mobile device ( $M$ ).

*Proof.* An attacker can attempt to modify a message  $\{(ID_m||Y||C), T\}$  into  $\{(ID_m||Y^*||C^*), T^*\}$ , where  $T^*$  is the attacker's current timestamp. However, such a modification will fail in Step 3, because an attacker has no way of obtaining the valid value  $Z = (ID_b^2)^{k \cdot s_m \cdot T} \bmod N$  to compute the valid parameter  $C$ . Furthermore, it is infeasible that an attacker can get  $Z$  using  $C = h(Z, Y)$ , because it is a one-way property of a secure one-way hash function by Definition 1. Thus, the impersonation attack cannot be successful.

## 6 Conclusion

In this paper, we have demonstrated that Hwang et al.'s scheme is insecure against impersonation attacks. An improved version is then proposed to defeat impersonation attacks. The improved user identification scheme based on ID-based cryptosystem is designed to repair the security weakness inherent in Hwang et al.'s scheme.

## Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments in improving our manuscript. This research was supported by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Assessment).

## References

1. Shamir, A.: Identity Based Cryptosystems & Signature Schemes. Advances in Cryptology CRYPTO'84. Lecture Notes-Computer Science. (1984) 47-53
2. Maurer, U.M., Yacobi, Y.: A Non-interactive Public-key Distribution System. Designs, Codes and Cryptography. Vol. 9. No. 3. (1996) 305-316
3. Tseng, Y.M., Jan, J.K.: ID-based Cryptographic Schemes Using a Non-interactive Public-key Distribution System. Proceedings of the 14th Annual Computer Security Applications Conference (IEEE ACSAC98) Phoenix Arizona. (1998) 237-243
4. Hwang, M.S., Lo, J.W., Lin, S.C.: An Efficient User Identification Scheme Based on ID-based Cryptosystem. Computer Standards & Interfaces. Vol. 26. No. 6. (2004) 565-569
5. Menezes, A.J., Oorschot, P.C., Vanstone, S.A.: Handbook of Applied Cryptograph. CRC Press. New York. (1997)

# The Wrong Challenge of ‘Pervasive Computing’: The Paradigm of Sensor-Based Context-Awareness

Eric Angaman

University of Jyväskylä,  
Faculty of Information Technology,  
Department of Mathematical Information Technology,  
Mattilanniemi Building Agora,  
P.O. Box 35, FIN-40351 Jyväskylä, Finland  
Tel.: +358 14 260 1211, Fax: +358 14 260 2209  
[erangama@cc.jyu.fi](mailto:erangama@cc.jyu.fi)

**Abstract.** This paper is concerned with providing a critical reflexion about Weiser’s vision implementation of what is supposed to appear as the new era of human computer interaction: Ubiquitous computing. The aim of this paper is not a review of the concept of Pervasive computing which is commonly, and at its advantage called Ubiquitous computing. Nor an unarguable inventory of contentions in favor of Pervasiveness or not. On contrary, this paper acknowledges a certain idea of inobtrusive interaction mobility but emits reserves of the means for realizing this vision. This paper encourages and challenges modestly the mobile computing community for adopting and reformulating new research directions for supporting Weiser’s vision.

## 1 Introduction

For most of us, interacting with computers has been mainly stationary. This is now changing in a big way. In fact, and on a different scale computation is moving beyond personal devices. With the advent of ever smaller, lighter, faster artifacts along with various sensing technologies, computation becomes embedded in everyday devices and free us of the single desk. Also wireless technology allows devices to be fully interconnected with the electronic world. The technology mentioned above offer many challenges for designers. In reality, these technologies move the site and style of interaction beyond the desktop and into a larger world where we live and act. If for designers, the desktop appears to be well-understood, the real world, however is complex and dynamic. The design challenge, then is to make computation useful in various situations that can be encountered in the real world. Then appears another difficulty : the issue of context or context-aware computing which is utterly controversial for a number of reseachers [1996], [1998]. What is a technology which is context-aware? Discussion of context and context-awareness suffers from the generality of the concept. A context-aware application is basically a technology which is able to sense the



situation in which it is immersed and adjust its actions appropriately. When humans talk, they use a wide range of mimics (e.g body languages) to express their ideas according to the spatial environment they are located. Both gestual and physical location have for goal and advantage to increase their 'conversational bandwidth' [2000]. Unfortunately, this human to human interaction does not apply quite well when humans interact with computers. By improving access to contextualisation, designers will make communication much more efficient in human-computer interaction and finally produce more useful computational services. If the goal of context-aware computing is to make human-computer interaction easier, then how context should be provided to computers? How to better take advantage of the physical space to render the human-computer interaction more efficient and intelligible? Additionally, the increase of mobility creates situations where the user's context, such as the location of a user and the people and objects around him, is more dynamic. As a natural advancement of wireless telecommunications capabilities, open networks, and continued increases in computing power have given the users the expectation that they can access information services whenever and wherever they are.

This paper addresses the quality of the reliability of sensor-based context recognition through the wireless spectrum in the effective use of human-computer interaction.

The first section describes the contribution of sensor acquisition to context-awareness. To gain context information the usage of sensors is proposed. The description of this contribution is extended and classified according to its respective application domain. This includes an analysis and discussion of the technical feasibility of context acquisition in a general context of computing mobility.

The second section considers the field of inquiry of the notion of sensing context and analyses commentaries that this reflection has raised in the literature. Finally, from a technical perspective a representative sample of important issues is enumerated and discussed. A general motivation and recurrent question underlying this paper is how Weiser's vision (e.g the ideal case) could reasonably be implemented by means of current technologies.

## 2 The Contribution of Sensor Acquisition to Context-Awareness

The actual utility of context-awareness in mobile systems has been demonstrated in a wide range of application examples [2000], [1999], as well as in indoors [2002], [2005] and outdoors [2002], [2001] environment. If context is the rationale, what constitutes context and how context could be captured, processed, and exploited for providing sustained added value and positive experiences to the end-user?

In this section, we explore several procedures for sensing the environment. Towards this goal, we illustrate context acquisition through relatives applications. Finally, issues and contribution of the sensor technology relevant to context acquisition are discussed.

## 2.1 Context Processing

### Context-Awareness

*Predicting and Approximating.* The use of sensor<sup>1</sup> data for context implementation consists of predicting or anticipating and finally approximating situations resulting from real world situations with respective consideration of context categorization and available sensing technologies.

*The Ideas of Capturing Data.* Sensing devices allow systems to have information about real world context data such as location of people, things and other artifacts in the spatial environment. In order to offer additional functionalities and to improve HCI<sup>2</sup> for mobile devices and appliances, research and literature have considered three major areas of interest to support these needs:

1. Sensor infrastructure

This approach (e.g indirect awareness [2002]) allows the dissemination of multitude of sensors spread across a geographical area for obtaining and providing context to smart devices. Practical examples of such technology are various. Wireless ad hoc sensor network figures among this technology.

2. Sensor integration

A more recent approach has fostered the idea that diverse sensors and several extraction techniques should be merged to alleviate the problems caused by complex algorithms and expensive hardware to simple sensors. This approach relies more on mobile terminal capabilities rather than on infrastructure. Examples of such technology (e.g the Smart-Its [2001]) are few as regard to their indirect counterpart.

3. Derived context

A modern approach (e.g derived context) has consisting of involving applications which make deductions from contextual data.

### Cue Generation

*The Machine Learning Mechanism.* Appropriate sensor selection coupling with a typical machine learning mechanism offer a positive approach for extracting and deriving raw from sensor data. Conceptually, the signal detection method and the cue extraction algorithms, employed in a situational context, are similar to any A/D Converter<sup>3</sup> used in Digital Signal Processing. As a result, the relationship between cues and context help to forecast plausible scenarios resulting from cues' description, and finally allowing artifacts for taking appropriate actions according to the user's location.

*Cue Attribute Extractions.* Context appears to be more closely related to the person's or the object's location whereas cues are assumed to be more generic.

---

<sup>1</sup> A sensor is a device that produces a measurable response to a change in a physical condition such as temperature or in a chemical condition such as concentration.

<sup>2</sup> Human-Computer Interaction.

<sup>3</sup> Analog-to-Digital Converter used to convert an analog signal into a digital format.

Towards this goal, the TEA<sup>4</sup> project [1998] has contributed in understanding of how multi-sensor context-awareness can be applied in mobile computing for the benefit of capturing context. The outcome of the research activity suggested that in a practical context recognition, results based on a single sensor may in many cases be unreliable. Realibility of recognition can be enhanced by fusing the features from many different sources [2003]. Analysis of this contribution to sensor based perception of real-world scenarios corroborates several aspects of perception findings in TEA project:

- Perception findings
  1. Useful context appears to be beyond location. As a result, additional information could be derived from it.
  2. Single powerful sensor architecture emphasizes location (e.g position) because it captures only restrictive aspects of environment. Multiple sensors architecture (e.g collection of diverse simple sensors) emphasizes context because it takes advantage of various specific aspects of the environment which gives an overall picture that better characterizes situation (e.g context) than location.
  3. On contrary of sensor fusion (e.g diverse sensors), the fusion of diverse sensors, as an approach to derive context improves perception. The goal is to improve accuracy and precision by integrating multiple type of similar sensors.

*From Smart Sensor Objects to Wireless Sensor Networks.* The basic and primary used of sensors was consisting in embedding sensors in the object itself, along with memory, power supply and wireless networking capability. The object then can communicate details about how it is being used. By continuously connecting a variety of sensors through short-range wireless connections, sensors networks take sensor technology beyond the capabilities of individual smart objects to an integrated component within a larger system. As a result, each sensor has wireless communication capability and some level of intelligence for signal processing and networking of the data.

## 2.2 Contribution of Sensing Technology to Location Aware Computing

### Location: A Key Part of Context-Awareness

*The Usefulness of Location Sevices.* Determining a mobile user’s current location with accuracy will be one of the most important functions of future mobile computing environments. Besides, many different technologies exist that provide location for both outdoor and indoor environments [2002], [2001]. Location

---

<sup>4</sup> Technology for Enabling Awareness is a joint project of the University of Karlsruhe, Nokia Mobile Phones, Starlab N.V. and Omega Generation funded under ESPRIT programme of the European Commission’s Fourth Framework as part of the call Esprit for IT Mobility. For further information see <http://tea.starlab.net/>.

information is a critical attribute of context information for context-aware applications used by mobile users. Location services take many forms and provide value in a diversity of ways, but the common denominator is the spatial data handling capability that links location to other types of data. Simply knowing where a mobile user is, or how far the mobile user is from someone or something, is typically not valuable by itself. Relating location to other pertinent information gives its meaning and value. This information is eventually used to provision the user more valuable services based on the user profile such as: personalized news, guidance services, and other relevant services.

Broadly, one of the greatest potential for sensor-based context-awareness consists of enhancing the user experience by minimising the active user effort needed in managing the local environment.

We gave an analysis of available sensing technologies for acquiring information from the user and the environment to facilitate HCI. We have also enlightened sensor capabilities perception techniques used for capturing environment. Moreover, discussions of sensor integration have shown that distributed sensors is an exemplary approach to capture real world situations and to fairly adjust HCI. Equally important, experimental results have showed that the concept provides a technical viable solution for providing significant benefits for user of mobile appliances. Context is fine in theory. Still, sensor integration with localization services for larger scales remains an open issue. Research on sensor-based context-awareness is purely theoretical and does not address the difficulties of implementation and evaluation in real use. As a result, a number of challenges still need to be addressed.

### 3 The Resulting 'Pervasiveness Controversy': Towards a Sensor Revolution?

As a solution for enhancing the user experience, multi-sensor awareness technology envisions a world in which almost every object will be able to sense, reason, communicate and act, adapted to the relevant context. From this point of view, Pervasiveness and Ubiquitous computing do not escape from this proliferation of sensors in our everyday lives. However, research activities have suggested that large networks of interconnected smart sensors should be implemented to better monitor and control our world for realizing Weiser's vision. As a prerequisite towards a sensor revolution in a pervasive environment, it belongs to the research community to wonder how to collate and correlate the sensor information and provide it in a useful way to any smart device.

#### 3.1 The Controversy

One of the main leaders in the controversial issue raised not by the concept itself, rather by its application is Thomas Erickson<sup>5</sup>. Erickson wishes that applications could adjust their actions appropriately in the context in which they

---

<sup>5</sup> A research staff member at IBM T.J. Watson Research Center, Hawthorne, NY.

are immersed. In fact, he feels reserved about the implementation success of such technology that he calls 'metaphor'. [2002]

## Issues

*The Controversy.* In fact, Erickson considers that we, as human beings integrate a broad range of values such as habits, manners, behaviors, dispositions, capacities and so on. The addition of all these qualities are empirically what he calls common sense. From this point of view, mankind with common sense would notice when being at the opera or theater that he should turn his handset in silent mode for instance. He does not deny that the application will be able to detect the motionless location of the user, nor the dark place with ambient noise. Consequently and by comparison, he deduces that the context-awareness of the human being and the one of any application are very different from each other. One is qualitatively characterized by empirism whereas the other is quantitatively characterized by sensors which have moreover the ability to detect and respond to a small range of features of their surroundings.

*His Arguments.* He really feels sceptical about the implementation success of such technology as regard as to the following reason :

- The prevailing argument consists of the ability of any application to be able to recognize the context and take appropriate action. According to him, this capability requires 'considerable intelligence'. He considers that if one day, the human being is able to implement a robust context-aware application which will rarely fail, that necessarily supposes the addition of new rules. We all know, based on experience that new set of rules increases complexity. The ultimate consequence is to make the application more difficult to understand. Interacting in a new and unfamiliar manner makes the interaction inaccessible and complicated. It seems that mankind has not learnt yet from his previous experiences. The case of Artificial Intelligence is a good example in its attempt of implementing complex systems or applications that require common sense. Current attempts in this direction have led to infuriate applications which fairly mimic mankind in a clumsy way. Nevertheless, from an era of needless technology, AI<sup>6</sup> seems to head for a more helpful technology. As a result, we should not be misleading by the goals of the two entities. This technology should not be used in this direction.

Computational systems are good at gathering and aggregating data; humans are good at recognizing contexts and determining what is appropriate. [2002]

## 3.2 Context Sensing

**Key Issues and Challenges Associated with Sensors.** Although there are many examples of smart sensor objects technologies achieving technical

<sup>6</sup> Artificial Intelligence.

goals [2001], [2001]. There are also too many examples of smart sensor devices development projects not achieving technical goals [2003], [2001]. However difficulties arise when creating a sensor network:

*Scalability.* Large number of mostly stationary sensors are necessary to capture the dynamic and rich semantic of any physical environment. To provide context-awareness to smart devices, networks of 10,000 or even 100,000 nodes are envisioned, so scalability is a major issue. So far, no densely sensor networks have not been deployed yet in the real world. It likely happens in the near future and these for two main reasons:

1. No universally accepted standards exist to enable massive and inexpensive implementation of wireless sensor networks. This issue is related with intellectual property ownership among manufacturers. Several standards have been proposed and are currently under development including the IEEE 1451 series [1998] and many others [1999]. These approaches provide standards for connecting sensors and actuators to networks.
2. The second issue concerns major technology barriers that in some instances limit the technology implementation of network sensors. These issues are described below.

*Increased Volumes of Data.* Wireless sensor networks will result in an explosive increase in data flows as networks become more pervasive. This will in turn raise a series of issues about how to deal with all the data or cues [2002]. New data and storage management methods and techniques will be needed.

*RFID Embeddeness and Privacy Issues.* In an Ubiquitous computing system, all items of interest (e.g bag, purse) must be individually tagged. In fact, RFID enables electronic labeling and wireless identification of objects using radio frequency communications. Especially as the technology gets smaller and smaller, an embedded RFID tag becomes potentially hard to find and difficult to remove. Moreover, the embeddeness of RFID tag could broadcast information without the knowledge of the mobile user when passing a scanner. Therefore, the public's reaction to a network system that can track almost any artifact could possibly be an obstacle.

*Cost.* Factors empeding the adoption of sensors are cost. It is very important to justify the overall cost of the network. Due to mass production techniques, sensor cost has noticeably decreased. Still, the cost of a single sensor remains a very challenging issue when considering the deployment of distributed sensor nodes.

*Power Supply.* Sensor nodes are limited in power, computational capacities and available storage capacity. Since in many applications sensor nodes will be placed in a remote area, service of a node may not be possible. In this context, lifetime of a node may be determined by the battery life, thereby requiring the minimization of energy expenditure. Altogether, with the proposed size limitations (e.g tiny and integratable into the environment), battery power alone does not suffice to ensure self-containment.

*New Applications.* Despite all the shortcomings mentioned above, doubtless that technologies for distributed sensor network will generate new breakthroughs.

1. Sensor will be smaller and more cost effective as a module.
2. Wireless technologies speed will increase with low latency and integrated in more reliable and larger network infrastructure.
3. Altogether, new competences and new skills will emerge for implementing and managing the sensor technology.

We have described real wireless applications of smart sensor device technologies with significant and economic feasibility concerns. The sum of issues makes sensor a fairly ordinary medium with the result that the sensor revolution creates more shortcomings to overcome than profitable solutions for end users. It is assumed that with the use of many sensors and the combination of different kinds of sensors (e.g sensor fusion) a large degree of awareness could be achieved. Still, experimental results have shown that sensor technology partly fails to capture some level of details inherent to the physical space. Sensor fusion captures only limited awareness information. [2001]

## 4 Conclusion

The broad aim of this paper was to argue the challenging and following question: 'How mobility while balancing with embedded computing can and should be integrated to capture the rich semantic of the physical space in such way that both permeate our environment?' Obviously, no absolute and definitive answer could be stated at this early implementation stage of pervasiveness. Each new technology provides increased benefits. Certainly, any new technology offers the potential to make life easier and more enjoyable. So do Pervasive computing and Ubiquitous computing, in their respective vision to improve the quality of life by creating the desired atmosphere and functionality via intelligent, personalized interconnected systems and services. As noted above, the implementation technology raises more issues than solutions.

For instance, one important issue for context-aware applications is that they must model the physical environment through an appropriate location model. However, no device is able to capture the rich, dynamic and semantic context due to its heterogeneous aspect. Mobile appliances can only recognize some aspects of context. To achieve true context awareness, mobile artifacts need to produce infallible real-world context data in the presence of uncertainty irrespective of location. The challenge, then for contextualisation applications will be to determine which parts of available context resources are relevant.

However many obstacles stand in the way before we can know where an object is all the time using sensor technologies due to the inadequacy of the tracking performance. Therefore creating an efficient and accurate sensing positioning system for indoor environment is a challenging task.

At the last, there is somekind of paradox in which sensor technology might be desirable from the perspective of network interface features, yet the economic

and technical viability may make sensors undesirable. Sensor-based context-awareness is limited by the technology available. Hence, to deal with the challenges outlined above, realization of Weiser's vision will require:

1. Investigating innovative sensing approaches using various sensors and algorithms.
2. To mature the theory of distributed sensor network systems enough (see section 3.2), further research is needed.
3. Bringing a large number of disciplines together.

Only when we have achieved this degree of interoperability will Weiser's vision become a reality.

Nevertheless, the future will be collaborative...

## Acknowledgements

The author would like to acknowledge Thomas Erickson, Vagan Terzian for their respective encouragements and benevolence. The author would also like to thank Tuomo Rossi, Mikko Jäkälä, Nazmun Nahar, Helen Kaikova for their valuable comments on earlier drafts of this paper. Some of the ideas expressed on this paper have benefited from discussions with Jani Kurhinen, Pekka Neittaanmäki, Timmo Männikö, Marko Mäkelä and Ismo Horppu. Finally, the author acknowledges a grant from the head of the Department of Mathematical Information Technology of the University of Jyväskylä: Timo Hämäläinen.

## References

- [2002] Erickson, T.: Some Problems with the Notion of Context-Aware Computing Communications of the ACM, **45**, 2, (2002) 102–104
- [2002] Gellersen H. W., Schmidt A., Beigl M.: Multi-sensor context-awareness in mobile devices and smart artifacts, Mobile Networks and Applications, **7**, 5, (2002) 341–351
- [1998] IEEE Std 1451.2-1997, IEEE Standard for a Smart Transducer Interface for Sensors and Actuators - Transducer to Microprocessor Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats, IEEE Instrumentation and Measurement Society, TC-9 Committee on Sensor Technology, IEEE, N.Y., Sept. 1998
- [2001] Hightower J., Boriello G.: Location Systems for Ubiquitous Computing, IEEE Computer, **34**, 8, (2001) 57–66
- [2002] Pahlavan K., Li X., Makela J.-P.: Indoor Geolocation Science and Technology, IEEE Communications Magazine, **40**, 2, (2002) 112–118
- [2005] Becker C., Dürr F.: On location models for ubiquitous computing, Communications of the ACM, **9**, 1, (2005) 20–31
- [2002] Peddemors A. J. H., Lankhorst M. M., de Heer J.: Combining presence, location and instant messaging in a context-aware mobile application framework, GigaMo-bile/D2.8 (TI/RS/2002/068) Telematica Instituut Enschede (2002) see [https://doc.telin.nl/dscgi/ds.py/Get/File-21982/PLIM\\_d28.pdf](https://doc.telin.nl/dscgi/ds.py/Get/File-21982/PLIM_d28.pdf)



- [2001] Davies N., Cheverst K., Mitchell K., Efrat A.: Using and Determining Location in a Context-Sensitive Tour Guide, *IEEE Computer* **34**, 8, (2001) 1–6
- [1998] Esprit Project 26900, TEA, (1998), see <http://tea.starlab.org/>
- [2001] Spirito M. A.: On the Accuracy of Cellular Mobile Station Location Estimation, *IEEE Transactions on Vehicular Technology*, **50**, 3, (2001)
- [2001] Beigl M., Gellersen H. W., Schmidt A.: Mediacups: Experience with Design and Use of Computer-Augmented Everyday Objects, Elsevier, *Computer Networks* **35**, 4, (2001) 401–409
- [1999] Schmidt A., Beigl M., Gellersen H. W.: There is more to context than location, *Computer and Graphics* **23**, 6, (1999) 893–901
- [2003] Korpipää P.: Bayesian approach to sensor-based context awareness, *Personal and Ubiquitous Computing*, **7**, 2, (2003) 113–124
- [1999] Crovella R. M.: *Sensor Networks and Communications, The Measurement, Instrumentation, and Sensors Handbook*, ed. Webster, John G., CRC Press/ IEEE Press, Boca Raton, FL (1999)
- [2002] Hjelm J.: *Creating Location Services for the Wireless Web*, ISBN 0-47140261-3, John Wiley and Sons, (2002)
- [1998] Dey, A. K., Abowd D.: Context Aware Computing : The CyberDesk Project. In: AAAI 1998 Proceedings Spring Symposium on Intelligent environments, (1998) 3–13. Technical report SS-98-02
- [2001] O'Connell, T., Jensen, P., Dey, A., Abowd, G. D.: Location in the Aware Home. Position paper for Workshop on Location Modeling for Ubiquitous Computing, In: proceedings UbiComp 2001 conference, Atlanta, GA (2001)
- [2001] Holmquist, L. E., Mattern, F., Schiele, B., Alahuhta, P., Beigl, M., Gellersen, H. W.: Smart-Its friends: A technique for users to easily establish connections between smart artefacts, In: Proceedings of UBICOMP 2001, Atlanta, GA (2001)
- [2003] Guvenc, I., Abdallah, C. T., Jordan, R., O. Dedeoglu: Enhancements to RSS Based Indoor Tracking Systems Using Kalman Filters, Accepted, GSPx In: proceedings of the International Signal Processing Conference, Dallas, TX, (2003)
- [2000] Morse, D. R., Armstrong, S., Dey A. K.: The what, who, where, when, why and how of context-awareness In: proceedings of the conference on Human Factors in Computing Systems, CHI '00 extended abstracts on Human factors in computing systems, The Hague, The Netherlands (2000) 371–371
- [2000] Farringdon, J., Oni, V.: Visual Augmented Memory (VAM) In: proceedings of the IEEE International Symposium on Wearable Computing (ISWC'00), Atlanta, GA, USA, (2000) 167–168
- [1996] Brown, P. J.: The Stick-e Document: A Framework For Creating Context-aware Applications In: proceedings of the Electronic Publishing, Laxenburg, Austria, IFIP,(1996) 259–272
- [2001] Mäntyjärvi J, Himberg J, Korpipää P, Mannila H: Extracting the Context of a mobile device user In: Symposium on Analysis, Design, and Evaluation of Human-Machine Systems (IFAC/IFIP/IFORS/IEA),(2001) 445–450

# An Abstract Model for Incentive-Enhanced Trust in P2P Networks

Mats Neovius

Department of Computer Science, Åbo Akademi University,  
Lemminkäisenkatu 14, FIN-20520 Turku, Finland  
mats@neovius.com

**Abstract.** Peer-to-Peer (P2P) networks have emerged as a prime research topic, partly due to the vast unexploited possibilities unrestricted distribution of the workload provides. The main hindrance for unrestricted exploitation of the P2P topology is, due to lack of security-related issues, the gullible attitude taken towards unknown agents. Therefore, the severity of the vulnerabilities caused by gullibility must be mended by other means, for example, by an effective incentive scheme encouraging agents to trustworthy behaviour. This paper presents an abstract model for incentive enhanced trust, to progressively assign the participating agents rights for accessing distributed resources, emphasising consistent behaviour. The model consists of a degrading formula, an illustrative incentive triangle and a best-effort distributed supervision model. Moreover, the same incentive model facilitates anticipation of future behaviour concerning any given agent founded on several distinct agents' opinion, suggesting that any knowledge concerning the counterpart is better than none.

**Keywords:** Peer-to-Peer networks, incentive, trustworthiness, anticipation.

## 1 Introduction

Reputation-based trust systems are widely studied and are probably the most realistic approach to anticipate future behaviour of an agent. Consequently, as in reality, there must exist a powerful incentive encouraging participants in a P2P network to credibly exchange information and act consistently benevolently. Thus, as mentioned by Kamvar, Schlosser and Garcia-Molina in [1], the identification must be a long-term user-specific, not relying on an externally assigned identity such as the IP address.

One way to encourage consistent behaviour is by assigning a covetous benefit to agents behaving benevolently. This gain should play the role of real-life money; it should be desirable and entitle to additional privileges. However, such an advantage attracts fraud in various forms. To describe the problems, it is essential to declare the basic frameworks and concepts which trust, in this case, is to be applied on.

### 1.1 Peer-to-Peer Networks

A P2P system implementing trust resembles inter-human communication in many ways. In a P2P network, all participating agents act as clients as well as servers and

possess equal rights, which suggest to a self-policing structure. Therefore, the definition concerning P2P architecture to be used throughout this paper is as follows:

A P2P architecture is a distributed network where each node grants other requesting nodes access to its shared resource(s). The resource(s) is/are accessible by any other participant on the network directly without intermediate servers.

Consequently, this paper views the participants in a P2P network as “members of a society”, where an agent’s actions are egocentrically determined by benefit. Moreover, a P2P system should be capable of handling any arbitrary agent’s unexpected drop-off from the network at any given time, without the network suffering any loss of service [2]. This excludes implementation of a predefined structure, such as servers or pre-shared secrets.

Despite the exclusion of central units, we argue that deploying an incentive in an agent-centric P2P architecture, a structured overlay network is a necessity. This is motivated because it enables systematic knowledge lookup, efficient collaboration between the participating agents for maintaining the incentive and assignment of credit to the appropriate agent. This paper considers the overlay system organised as a Pastry Distributed Hash Table (DHT) architecture. The Pastry DHT system provides scalability, low network diameter and proximity selection [3, 4].

## 1.2 The Trust Metric

Trust is a social phenomenon and can only exist between two distinct matters of which at least one is capable of feeling. As such, all models of trust should be based on the same as the social trust, knowledge about the counterpart. This paper discusses unrestricted agent-centric trust and situations where it is assumed that the counterpart is behaving irrationally. Walsh and Siner in [5] propose an object centric reputation scheme that is restricted to a specific kind of objects, in their case files. Such a system is however, unsuitable for agent-centric reputation evaluation because peers’ behaviour vary.

Implementing trust to be processed in a microprocessor requires that it can be measured and thereby, compel assigning a value for the metric. In a binary formation, an agent is evaluated as either trustworthy (affirmative) or untrustworthy (negative). Eventually every assessment should fit the binary formation. Considering the perpetually changing environment and variety of levels demanded, binary formation is insufficient for comprehensive usage. Therefore, the trust metric is considered in this paper discrete, between 0 (none) and 1 (complete), with a sufficient amount of states. According to calculations made on the values, the trustor will assign the trustee-specific rights to access and/or exploit resource(s), as will the trustee select the provider.

Besides the value of the metric, it must be distinguishable on a per actor basis and thus, explicitly mapped to a unique ID, as humans are recognised by characteristics such as the voice, by sight etc. Consequently, we argue that a unique ID is a precondition for implementing trust of any kind between conditionally trustworthy matters.

### 1.3 Recognised Abuses in P2P Networks

The present gullible approach adapted by participants in a P2P network, and the limited possibilities to locate colluding agents, attracts abuses of many kinds. Concerning peer misbehaviours, three types are recognised: collude inflation, deflating and faking [6].

Collude inflation are situations where a conglomerate of agents collaborate by reporting positively about each other in order to achieve a higher trustworthiness value. The problem is present in centralised online auctions and in reality, because there is no way to verify the feedback's truthfulness and dignity. However, including only one report per agent such as in eBay.com [7] and degrading the information by time would hamper any colluding intentions.

Deflation is a situation where a set of agents defames another's reputation by reporting unsatisfactory behaviour concerning it. This is comparable to spreading rumours in reality. However, degrading of reports as for collusion, affects deflation equally and is a feasible countermeasure.

A faker is an agent that introduces itself as another agent (usually) possessing a higher reputation. This problem should be solved at the assignment of the ID or at the mapping of the feedbacks to an ID. However, this is out of the scope of the topic and this paper assumes that the ID's are unique and the feedbacks are authentic.

Besides the misbehaviours concerning reputations; annoyances such as "free-riding" and "tragedy of the commons" are widely acknowledged. Both are consequences of unfair exploitation and contribution respectively, of the commonly accessible resources and can be solved utilising the kind of incentive presented later in this paper.

## 2 Trust in an Open Environment

A trust relation can be of many forms; it can be one-to-one, many-to-one, or many-to-many [8]. Optimally, the relation is many-to-one, where the knowledge about the counterpart is based on a combination of several sources' experiences. However, a distinction of the knowledge credibility according to sources' trustworthiness is required. This paper considers a three-level hierarchy of knowledge sources: a personal opinion, trusted agents' opinions and a public opinion.

### 2.1 Personal, Trusted and Public Opinions

As when considering humans, trust between P2P-networked agents should equally count on the capability of distinguishing between trust derived from different matters and events. Deducing personal opinions based on personal experiences is essential. However, in some situations the observations cannot cover adequate knowledge and relying on others' judgements is necessary. The trusted agents' opinions are "advises" and acquired gathering information by inquiring friendly sources. A public opinion is one reflecting the majority's opinion concerning the matter. Hence, the personal opinion is a concern that is alterable only by the possessing agent. Moreover, each agent should contribute in providing and maintaining a public opinion and collaborate with personally trusted agents to enforce understanding about the counterpart, which

is/are considered advised and trusted conditionally. Thereby, the agents form sets of reciprocal trusted conglomerates.

The public opinion does not alone solve any of the problems mentioned. It should be considered as we consider, for example, reviews at [epinions.com](http://epinions.com) [9]. Therefore, the public opinion can, at most, mend the uncertainty left by the trusted and personal opinion. However, uncertainty should have primary influence on the decision concerning selecting the agent to process the event. This is motivated by the threat of a colluding set of malicious agents collaborating in building a benevolent public opinion.

Because of the anonymity and egocentric behaviour, it is justified that the personal opinion has greater influence on the final outcome than the trusted agents and public opinions [10]. Consequently, a hierarchy of credibility is formed where the personal opinion mends with the trusted and only then with the public opinion. This hierarchy severely hampers the effect of collusion and deflation. However, the different levels of opinions must not result in conflicts though possibly indicating an opposite outcome, and a method of achieving a consensus is needed. This consensus should handle situations such as, for example, when the public opinion suggests negative assessment while the trusted and personal opinion suggest to affirmative with some uncertainty. In addition, a consensus method of the opinions adjusts the personal opinion to the trusted and public opinions and reduces “obstinacy”. The consensus of the different opinions results in a situation equal to inter-human interaction, i.e. that a maliciously behaving agent is capable of taking advantage of the conglomerate of reciprocally trusted agents’ benevolence only a finite amount of times.

## 2.2 Feedback Formation and Distribution

Trust relying on a public opinion in P2P networks is motivated because no single entity can have accurate information about all others’ conducts. Initially, no data concerning the counterpart exist, suggesting that reputation has to be built from some state. The state of no reputation, and thus the initial state, is considered in this paper as the state of uncertainty; because modelling trust in dynamic networks cannot allow confusion between “don’t trust” and “don’t know” [11].

A feedback is the generated data concerning the provider of the resource(s) after an event. The generated data is stored locally and submitted to the supervising agent including the ID of the counterpart, a timestamp, the feedback score and the ID of the reporter [12]. Additional application-specific data can be added. Including IDs in the feedback provides a possibility to identify and verify the transaction. In addition, the agents should monitor their reputation and when disagreeing on an evaluation, change the personal opinion about the reporting source accordingly. The timestamp enables utilisation of a degrading formula, with the justification that attitudes can change over time. The feedback itself is graded with a triplet of values; *belief* ( $b$ ), *disbelief* ( $d$ ) and *uncertainty* ( $u$ ). As discussed in section 1.2 and because the metrics are in contradiction and complete equal 1, their sum must equal 1.

Considering the definition of a P2P network, the feedbacks must be stored on the connected (live) agents. As a countermeasure for colluding inflation, the agents supervising feedbacks concerning any given agent should perpetually change. Moreover, the agent the feedbacks concern should not be included in the lookup chain

of locating the supervising agent. Therefore, the feedback supervising agents must be known by all participants all the time. Enabling this in a system utilising Pastry DHT is possible by having the trust supervising agents' IDs dynamically assigned by a hard-coded function in the application. This requires the DHT to assign the IDs dynamically on a per-session basis as a countermeasure for colluding alternation. However, the need of a unique static ID for each participant compels usage of two interconnected DHTs, each consisting of  $x$  tier to maintain scalability. In such a system, one layer provides the static *nodeID* while the other layer accounts for proximity selection, lookups and the feedback, being dynamically assigned, hereafter denoted *sessionID* (*sID*). This way the needs of a static unique ID and the requirements for countermeasures are satisfied.

Requiring any reporter to file the feedback to, for example, two closest supervising agents of its own *sID*, would provide data redundancy. That is, if  $sID\ c < d < e < f$ , the agent with *sID*  $e$  files reputation regarding *sID*  $c$  to *sID*  $d$  and *sID*  $f$ . A possible recovery can be conducted by a logical expression, where peer  $g, h, i, j$  and  $k$  represents adjacent supervisors for  $x$ , according to the distribution.  $i$ 's stored data can be retrieved by  $data \in h \vee j \wedge \neg g \wedge \neg k$ . In other words, if *sID*  $i$  fails, its data can be recovered by summarising all data that *sID*  $h$  or *sID*  $j$  store and that are not stored by *sID*  $g$ , nor by *sID*  $k$ .

Moreover, the redundancy provides a way for a newly assigned supervising agent to verify the passed feedbacks. In addition, such *sID* data passing provides means for semi-symmetrical distribution. Consequently, in order for colluding inflation to succeed, the malevolent agent should cooperate with the majority of the involved dynamically changing supervising agents. The feedbacks reported to the supervising agents are the values resulting in the public opinion that is a sum, calculated by a subjective logic, for example, the one presented in [13], of all feedbacks from a set of interactions with the agent(s) concerned.

### 3 The Incentive

In reality an incentive is very simple. It is usually money, fame or some other covetous benefit that good performance entitles to. However, distribution of the beneficial is complex. The incentive to be deployed for usage in computerised communication must be based on the idea of giving benefit to the active and benevolent agents and reducing the value of the beneficial as a consequence of unsatisfactory performance. As a result, there must exist a *carrot* as well as a *stick*. In order to increase the anticipations truthfulness, experiences should degrade according to time.

#### 3.1 A Degrading Formula for Trust

Philosophically, trust can never be absolute [14]. The core idea of this is the fact that even a friend, considered as trustworthy, can fail the expectations; respectively can an untrustworthy agent behave benevolently. To meet these challenges, a degrading formula must weaken the weighs of the feedbacks based on time and sociality. This is necessary in order to give less social agents equal possibilities; weakening recent

experiences less. Whitby, Jøsang and Indulska [15] proposed a formula without the sociality factor, however, including it in the same formula is easy, resulting in formula 1.

$$p_{Z,t_R}^{X,t} = \lambda^{t-t_R} \gamma^{(l-k)} p_{Z,t_R}^X \tag{1}$$

In formula 1,  $p_{Z,t_R}^{X,t}$  is agent X’s rating of agent Z at time  $t_R$ , t being the current time. In other words, an event occurred at time  $t_R$  where agent X rated agent Z, the current time being t.  $0 < \lambda < 1$  is the longevity factor degrading the rating according to time. The  $\gamma^{l-k}$  represents the ordering of the feedback by occurrence, l being the selection’s size and k the position number where the most recent is l, degrading according to sociality and being  $0 < \gamma < 1$ .

The formula should be applied upon the *belief* and *disbelief* values in personal opinions’ every experience. Because the sum of the metrics is 1, uncertainty equals 1-b-d. In addition, formula 1 sociality factor covers the claim that complete trust or distrust cannot exist, and is a countermeasure for key-space depletion, dropping agents with uncertainty exceeding some predefined threshold value. The values assigned for  $\lambda$  and  $\gamma$  are subject to the application and the environment. The  $\gamma$  value should adjust to the frequency of attitude changes; the lower value, the heavier weight on recent events.  $\lambda$  depends on the frequency of transactions conducted with the counterpart.  $\gamma$  and  $\lambda$  combination reacts to changes in attitude and allow the agent to adapt to the environment. Moreover, the degrading formula is forgivable and will grant the maliciously behaving agent a new chance, after a given time, depending on the longevity factor, of acquiring favouring among the reciprocal conglomerate it tried to fool.

### 3.2 Calculating with the Metrics

Calculating and enforcing the accuracy of the metrics is essential in order to reach the decision. Figure 1 illustrates a situation where two trusted agents, Bob and Claire, contribute in enforcing Alice’s anticipation concerning the target, David.

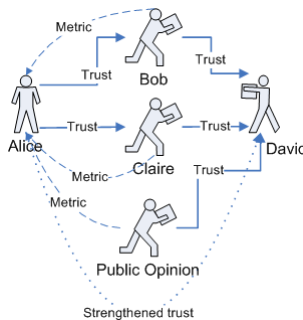


Fig. 1. Trust combination

The trusted agents participating in the evaluation should contribute with their personal opinions to the requesting entity, without enforcing their understanding by querying further or redirecting. This is motivated because Alice trusts Bob and Claire, not a fourth party, to evaluate David. A situation alike the one in Figure 1 compels a consensus to be achieved between Bob’s and Claire’s metrics. Bob’s and Claire’s consensus will eventually be combined with Alice’s personal opinion, and finally patched by the public opinion, resulting in the final opinion.

The calculation merging the participating agents’ degraded metrics is based on probability calculations and can be performed according to formula 2 illustrated below, originally proposed in [16].

$$\begin{aligned}
 \text{belief}_N^M &= (b^N * u^M + b^M * u^N) / (u^M + u^N - u^M * u^N) \\
 \text{disbelief}_N^M &= (d^N * u^M + d^M * u^N) / (u^M + u^N - u^M * u^N) \\
 \text{uncertainty}_N^M &= (u^N * u^M) / (u^M + u^N - u^M * u^N)
 \end{aligned}
 \tag{2}$$

*M* and *N* are any agents which personal opinion metrics are to be merged; in this case Bob and Claire. If several agents contribute, the merging is done between any two agents or sets of agents at the same level of the consensus process. Eventually the consensus will reach such magnitudes that it represents the understanding of the underlying group.

The final patching of the uncertainty for the expected outcome utilising the public opinion should be performed after applying the metrics from the trusted agents. This can take place utilising, for example, the following formulas.

$$\text{belief}_{\text{public}}^{\text{calculated}} = \text{belief}^{\text{calculated}} + \text{uncertainty}^{\text{calculated}} * \text{belief}^{\text{public}}
 \tag{3}$$

$$\text{disbelief}_{\text{public}}^{\text{calculated}} = \text{disbelief}^{\text{calculated}} + \text{uncertainty}^{\text{calculated}} * \text{disbelief}^{\text{public}}
 \tag{4}$$

In these formulas, *calculated* denotes the degraded trustworthiness of the levels higher in the hierarchy, acquired by formula 2 and 1. Mending this calculated opinion with the public opinion that does not recognise uncertainty, forms an opinion correlating to the expected outcome based on the available knowledge.

Utilising these methods, the trust metric fits the triangle illustrated in Figure 2, when uncertainty is included and the anticipation of forthcoming behaviour is possible. Thus, all possible providers of the requested service can be compared and the most suitable chosen.

### 3.3 An Incentive View

In every incentive method, the inducement must be such that the users cannot gain from reinitiating with a new identity [1]. Hence, we argue that the initial state must be equal to or worse than the state of untrustworthy, with the justification that any knowledge to base anticipation on reducing the risk of misjudgement is better than none. This results in the idea that the state of disbelief is preferred to the initial state, countermeasuring whitewashing.

This paper considers the initial state as the state of uncertainty, a state where no anticipation about future behaviour based on reputation is possible. At the same time,



the state of uncertainty indicates that the ID is available for any requesting newcomer. The incentive triangle, derived from the opinion triangle in [16], illustrated in Figure 2, summarises these ideas.

The triangle should be interpreted so that each vertex represents completeness. Therefore, the trustworthiness of any agent consisting of three metrics is representable by one point in the triangle. The median starting at each vertex is the grading of the different values, where belief is represented by  $Q$ , disbelief by  $R$  and uncertainty by  $P$ . The dot represents an example (personal opinion), with belief ( $Q$ ) 0.25, disbelief ( $R$ ) 0.65, uncertainty ( $P$ ) 0.1.  $E(x)$  presents the mending (expectation), illustrated in formula 3 and 4, where the personal opinion's uncertainty is mended by the public opinion, whose value is represented by the dotted line  $a_x$ . In Figure 2, this starts at uncertainty, ending at *belief* = 0.6 and thus *disbelief* = 0.4.

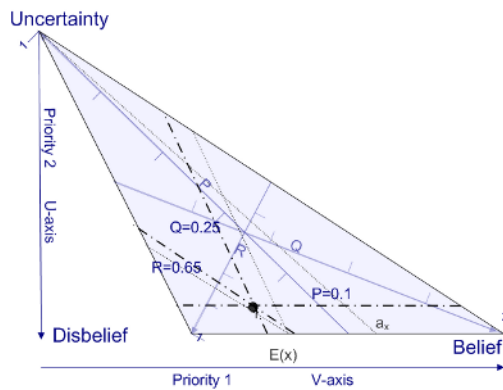


Fig. 2. Incentive triangle

When calculating the expectations value, the final value is required to be either affirmative or negative and thus uncertainty must equal 0. Uncertainty is reduced to equal 0 by applying the public opinion on formula 3 and 4, resulting in the removal of the uncertainty metric. The degrading formula 1 affects the opinion in the way that it moves towards uncertainty on the axis with the original relation between trust and distrust. Moreover, the triangle recognises two priorities, which are determined by trust qualities and thus purpose specific.

In this specific view, a newcomer is not assigned any profit, which should be the best countermeasure for avoiding an agent with bad reputation to reinitiate its trust relation in form of signing in with a different ID. This implies that the participants are encouraged to consistently act using the same identity every time.

The presented ideas maintain a balance between capability to operate and actual trustworthiness. If some agent is incapable of fulfilling the placed expectations, its trustworthiness will suffer among the expecting agents. Consequently, the network has reacted to this successfully and the trustworthiness/capability balance is maintained.

## 4 Conclusion

Combining the models presented in this paper reduces the presented problems' severity. Colluding inflation can occur a finite amount of times per conglomerate of reciprocally trustworthy agents because of the influence of the personal opinions. Deflation compromises the public opinion but the target maintains its ability to operate due to the personal opinions and will recover because of the degrading formula. Issuing countermeasures for faking is very difficult, if not impossible, without pre-shared secrets or intermediate authenticating servers. The "free-rider" and the "tragedy of the commons" problems are solved by a *carrot - stick* relation and utilisation of the personal and public opinion. In addition, the presented incentive reacts to changes in attitude and provides a possibility for malevolent/passive behaviour to change without re-identification.

The problems remaining are the evidence concerning a feedback and the assigning of a unique ID. These issues are of different character and we cannot see the way these could be solved utilising an incentive. Moreover, credentials are excluded from this paper, but being an extension of trust relationships, they are an essential part of trust in reality.

Any accurate simulations to enforce the claims in this paper are difficult to make because the contribution is in anticipation of the irrational. Simulations can thereby not reach greater accuracy than having a static value to calculate irrationality from, which is superficial. The reason is that this would imply simulating human behaviour, but since the human society is functional, creating a similar environment for computerised communication should be the objective. This paper has provided some ideas in order to reach this objective from the point of view that nature has evolved the ultimate trust formation scheme.

## References

1. Spandar D. Kamvar, Mario T. Schlosser, Hector Garcia-Molina (2003). "The EigenTrust algorithm for reputation management in P2P networks". In Proceedings of the Twelfth International World Wide Web Conference, May, 2003.
2. Rydiger Schollmeier (2002). "A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications". Proceedings of the First International Conference on Peer-to-Peer Computing (P2P'01).
3. Dan S. Wallach (2002). "A Survey of Peer-to-Peer Security Issues". International Symposium on Software Security (Tokyo, Japan), November 2002.
4. Miguel Castro, Peter Druschel, Y. Charlie Hu, Anthony Rowstron (2002). "Exploiting network proximity in peer-to-peer overlay networks". Technical Report MSR-TR-2002-82 (2002).
5. Kevin Walsh, Emin Gun Sirer (2005). "Thwarting P2P Pollution Using Object Reputation". Cornell University, Computer Science Department Technical Report TR2005-1980.
6. YangBin Tang, HuaiMin Wang, Wen Dou (2004) "Trust Based Incentive in P2P Network". Proceedings of the IEEE International Conference on E-Commerce Technology for Dynamic E-Business (CEC-East'04).

7. eBay.com. "Understanding feedback scores". URL: <http://pages.ebay.com/help/feedback/feedback-scores.html>
8. Tyrone Grandison, Morris Sloman (2000). "A survey of trust in internet applications". 4th Quarter 2000 issue of IEEE Communications Surveys & Tutorials.
9. Epinions.com. <http://www.epinions.com>
10. Vinny Cahill, Elizabeth Gray, Jean-Marc Seigneur, Christian D. Jensen, Yong Chen, Brian Shand, Nathan Dimmock, Andy Twigg, Jean Bacon, Colin English, Waleed Wagealla, Sotirios Terzis, Paddy Nixon, Giovanna di Marzo Serugendo, Ciarán Bryce, Marco Carbone, Karl Krukow, Mogens Nielsen (2003). "Using trust for secure collaboration in uncertain environments". IEEE pervasive computing, volume 2, number 3, July – September 2003, page 52 – 61.
11. Marco Carbone, Mogens Nielsen, Vladimiro Sassone (2003). "A formal model for trust in dynamic networks". BRICS Report RS-03-4, 2003.
12. Li Xiong, Ling Liu. "PeerTrust: Supporting reputation-based trust for peer-to-peer electronic communities". IEEE Transactions on knowledge and data engineering, vol. 16, no. 7, July 2004.
13. Audun Jøsang (2001). "A Logic for Uncertain Probabilities". International Journal of uncertainty, Fuzziness and Knowledge-Based Systems. 9(3), pp.279-311, June 2001.
14. Martin Hollis (1998). "Trust within reason". Cambridge, United Kingdom, Cambridge university press.
15. Andrew Whitby, Audun Jøsang, Jadwiga Indulska (2004). "Filtering Out Unfair Ratings in Bayesian Reputation Systems". In the Proceedings of the Workshop on Trust in Agent Societies, at the Third International Joint Conference on Autonomous Agents & Multi Agent Systems (AAMAS2004), New York, July 2004.
16. Audun Jøsang (1997). "Artificial Reasoning with Subjective Logic". In Proceedings of the Second Australian Workshop on Commonsense Reasoning, 1997.

# Construction of Credible Ubiquitous P2P Content Exchange Communities

Yuki Yokohata, Hiroshi Sunaga, and Hiroyuki Nakamura

NTT Network Service Systems Laboratories, NTT Corporation,  
3-9-11 Midori-cho, Musashino-shi,  
Tokyo 180-8585, Japan  
yokohata.yuki@lab.ntt.co.jp

**Abstract.** This paper describes methods for suppressing illegal behavior in P2P networks to construct ubiquitous P2P content exchange communities. Although rigid digital rights management has been established elsewhere, it requires rather a large processing load and time, and it is mainly effective at preventing illegal behavior of end users. Here, we propose a more efficient method for processing content distribution. It aims to control content exchange so that illegal activities can be reduced to a sufficiently low level. By observing content exchange interactions in rendezvous points, it detects illegal activities and identifies which peer performed them. Simulation results show the effectiveness of the proposal.

## 1 Introduction

We are now at the dawn of the era of a ubiquitous communication society, where our daily lives are becoming more and more dependent on various types of telecommunications. In this society, broadband IP networks will play a key role in supporting the social infrastructure as well as mobile communication networks. One of the most promising applications is content exchange, by which various types of content, created by professionals, semi-professionals, or amateurs, will be traded anywhere.

Moreover, a large majority of Internet users have experience in using or are even now using peer-to-peer (P2P)-based file exchange. Most P2P-based applications are illegal, and 60% of users feel anxious about illegal activities such as fraudulent electronic transactions, according to a report by the Ministry of Internal Affairs, and Communications (MIC), Japan [1]. However, P2P is technically promising and there must be potential users for legitimate content exchange services. In other words, if a legal community for content exchange is provided by a trusted service provider, or created only by trusted members, then legitimate content exchange markets will flourish.

As we can see from this background, various methods for achieving reliable operation of the community are necessary. Although there are various types of illegal behavior by members in a ubiquitous or P2P community, this paper focuses on how to obtain information about the status of content exchange and illegal behavior in the community, which will lead to stable and reliable operation and accelerated content-trade activities.

The rest of this paper is organized as follows. Section 2 clarifies problems and requirements for ubiquitous content exchange. In Section 3, we propose a new

framework for identifying the status of the community, in particular illegal behavior by members. Section 4 presents simulation results and analyses. Finally, in Section 5, we present conclusions and show future directions.

## 2 Problem Analysis and Requirements

### 2.1 Objectives of Ubiquitous Content Exchange Community

A ubiquitous content exchange community is made up of users, content creators, and operators. The creators may want to sell contents that they created either originally or by modifying the content of another creator. They may sometimes deliver content for free to demonstrate their talent. When other people use this content, they may want to clarify their own rights and conditions of use. Content may be original or modified music, pictures, or movies for mobile terminals, ordinary PCs, or home information appliances. Even content created by non-professionals has potential for trade. Users may want to obtain desired content, or value-added content. Of course, both users and creators want to communicate with those who can be trusted. The community operator mediates between them by providing a safe trading environment, which leads to a new business model. It is essential to observe the status of content exchange, to trace and exclude illegal members, and to manage charging or settlement in the community.

### 2.2 Problems to be Solved

We assume that the content exchange community takes a P2P-form and is connected to authorities for the identification of users and content. Figure 1 shows a typical sequence for content exchange in a P2P community. In this processing, there are several types of illegal activities that can occur, as explained below.

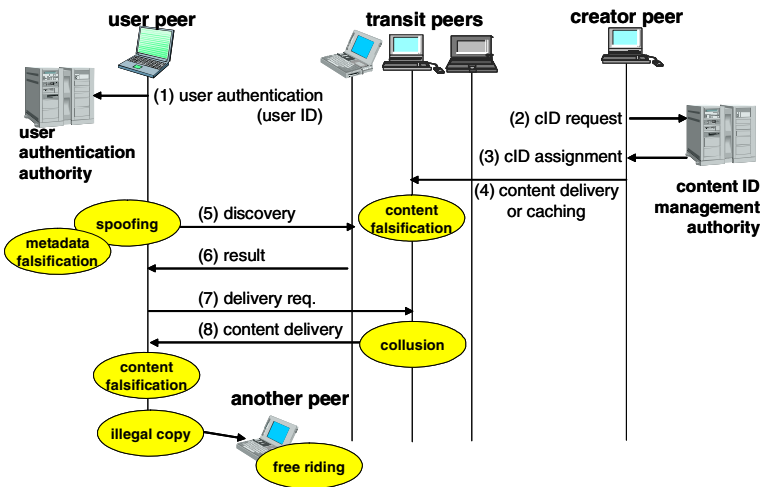


Fig. 1. This shows content exchange sequence and problems

- (a) Content falsification: content transferred through transit peers might be falsified and falsified content might be cached at a transit peer.
- (b) Spoofing: some peer might spoof its address (i.e., its identity) and gain access to content illegally.
- (c) Metadata falsification: during discovery processing, metadata might be falsified to make fake content that imitates copy-righted one.
- (d) Collusion: a peer might send paid-for content to a partner peer that is not eligible to receive it.
- (e) Illegal copying and free riding: a peer might deliver illegally obtained content to other peers, that is, free-riders.

### 2.3 Existing Countermeasures

There are three approaches to deal with these problems, i.e., prevention of illegal behavior, suppression of illegality, and permission to use content with some conditions. The most appropriate method depends on the value of content, the application type, the content holder's policy, or the operator's policy.

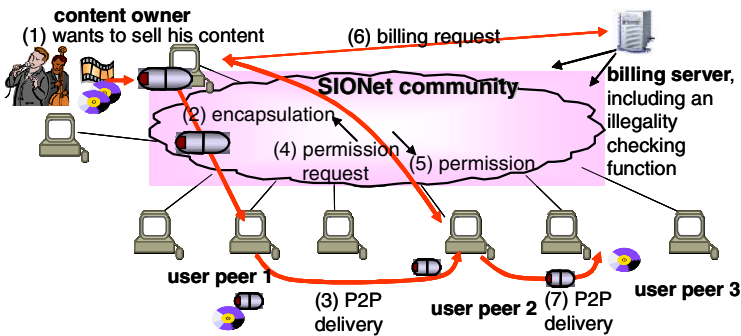


Fig. 2. This shows the P2P DRM model

#### 1. Prevention of illegal behavior

This is a rigid type of countermeasure that does not allow content to be viewed without permission. Figure 2 is a rigid approach to protect the copyright of content [2]. A content owner first encapsulates his content with necessary information such as the number of times that users may view it. This capsule is viewed only through a proprietary browser and cannot be opened without his permission. Then, the capsule is delivered to a requesting peer in the P2P mode, and this peer asks the owner for permission. If the user intends to pay for the content, the owner returns the permission. Upon receipt of the permission, the user can view the content. Until the specified time is reached, the content can be relayed to another peer.

This method is firstly applicable to countering illegal copying and free riding. Although it is also effective against content falsification, illegally obtained content can be exchanged by this rigid capsule. Spoofing and collusion are meaningless because the viewer must pay to view the content. It is not considered to be effective against

metadata falsification. Note that encapsulating content takes a long time and involves a heavy processing load.

## 2. Suppression of illegality

This method aims at managing or controlling content exchange so that illegal activities can be reduced to a sufficiently low level. In this paper, we mainly deal with this approach by using some mechanism implemented in content ID forum [3] from the P2P viewpoint.

## 3. Permission to use content with some conditions

This method, famous as 'Creative Commons' [4], allows users to freely view or modify content with some conditions. Eleven sets of conditions (rules) are defined according to the combination of licensing elements. Its aim is to use the flexibility of copyright law to help an active usage or secondary creation of various contents.

This approach has been implemented, for example, in the Digital Commons system [5][6].

# 3 Modeling the Suppression of Illegality

## 3.1 Identification of Community Elements

Our basic proposal is to trace content exchange interactions through the cooperation of agents residing in the community. To identify content, each of them is assigned a unique and secure identification, i.e., content ID (cID). This ID is authenticated by an authority. Also, each peer in the community is given a user ID by a user authentication authority. The authentication processing is guaranteed through the use of public key mechanisms. By using a metadata-matching-based discovery capability, as for example, KaZaA[7] does, peers or content can be found based on properties given in the form of metadata, such as MPEG-7[8].

## 3.2 Operational Policies

We decided upon the following operational policies to meet the objectives.

- The basic operation is to log cIDs and metadata exchanged in the community. Logged information can be used to judge whether or not the license, price, or usage conditions of content have been falsified.
- The use of a user ID prevents spoofing. The cID is linked with the hash value of the content to be registered to guarantee its uniqueness. Because the same cID is given to copies when the content is duplicated, illegal content is linked with the same cID.
- Metadata used in this community contains information about conditions for secondary use of the content (XML description) because this community aims to activate content creation by feeding back incentives to the creators.
- There are two methods for assigning an identifier to content: 1) combining header descriptions and digital signature and 2) inserting an electronic watermark. Although the latter has better performance against illegal actions, the operational cost is higher. This paper treats the former method.

Things for the creator who wants to protect his content to do are

1. getting the cID for content from the authority
2. calculating the hash value of the content
3. attaching the digital signature of cID, hash value and metadata to content
4. attaching metadata and cID to the header for searching

### 3.3 Proposed Processing Mechanism

The basic framework discussed thus far is to trace interactions of content exchange to detect peers performing illegal actions. Metadata, cID, hash vale that are attached during content delivery is logged somewhere in the community. If originating and destination peers log these, falsification is rather easy, which leads to incorrect logging or sabotaged logging. We position rendezvous point peer (RP) [9] to perform logging on the route between these peers, as shown in Figure 3.

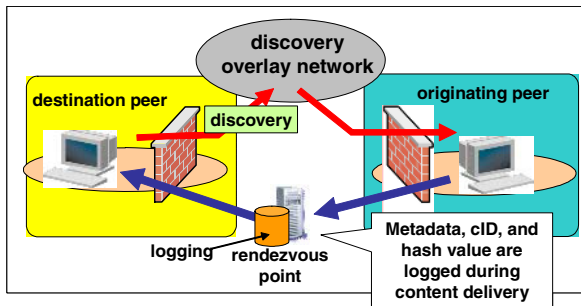


Fig. 3. This shows our proposed model of positioning rendezvous point

The obtained log data must be credible. To guarantee credibility, one peer selected by the community operator acts as the super RP peer (SRP). Its functions are to record the cID, metadata and hash value of content newly uploaded to this community and compare them with the data logged in the community. If a mismatch is detected, falsification is recognized. For example, if content is falsified, its hash value changes. A RP Peer detects the content falsification from this difference. If metadata is falsified, RP peer can detect the difference in metadata of capsules having the same content ID.

However, these SRP functions are too heavy to be performed by only one peer. Therefore, logging itself is performed in underlying RPs, managed by the SRP. RPs send logged data to the SRP at the time an ordinary peer starts content delivery. This mechanism is shown in Fig. 4.

To guarantee the credibility of the community, credible RPs must be selected.

RPs are selected from ordinary peers considering the following conditions.

1. Each candidate must have a global IP address.
2. Each candidate must have higher CPU performance and a larger memory size than ordinary peers.



3. Each candidate must leave the community less frequently than ordinary peers. Of course, these physical conditions alone cannot prevent illegal activities. Some RPs might make an illegal linkage between cID and content, or falsify content as well as logged data.

Therefore, credible RPs are selected based on the calculated trust value. We use an equation defined in [10]. After RPs have been selected randomly from those satisfying the physical conditions, they are narrowed down to more credible RPs. The SPR compares logged data sent from RPs with original data obtained directly from the content creator. The equation of the trust value of an RP ( $Tr(n)$ ) is

$$Tr(n) = a * Tr(n-1) + (1-a) * Sc \quad (n > 1). \tag{1}$$

Here, 'n' means the number of interactions for content exchange and 'a' is a constant. If a match is made, then 'Sc' is increased by 1, otherwise it is decreased by 1.

Then, we determine a threshold value to judge whether or not falsification has occurred. If  $Tr(n)$  is less than threshold  $h$ , this means that falsification has been detected in the RP. To make this equation more effective, the following condition is required.

$$2a - 1 < h \tag{2}$$

If  $Tr(n)$  is less than threshold  $h$  from the beginning, it means that an illegal RP has been selected. This RP is disqualified and another peer that has higher physical conditions is selected. Of course, at the time of selection, it is not known whether the new RP is credible. Later activities will be checked and used for judgment.

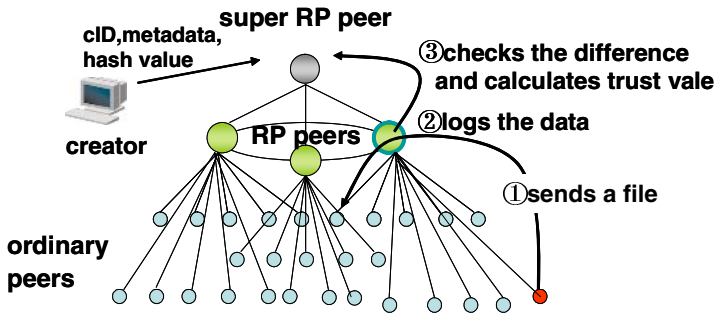


Fig. 4. This figure shows layered architecture of peers

## 4 Evaluation and Analyses

### 4.1 Simulation

We performed simulations to evaluate the proposed algorithm discussed in Section 3. In particular, we focused on the appropriate number of RPs in a community. Then the simulator computed the number of content-exchange interactions required for a newly selected RP to detect illegal peers in the community and to augment its trust value to the

credible level ( $=1$ ). This simulation identified how this value is affected by the peer configuration, i.e., the ratio of illegal actions to the total number of actions (illegal action rate) and the ratio of the number of RPs to the total number of peers in the community (RP rate). We assumed that the number of peers in the community was 1000, the total number of interactions was 10,000, and the parameters of equations (1) and (2) were fixed to  $a=0.85$  and  $h=0.75$  to raise the detection efficiency.

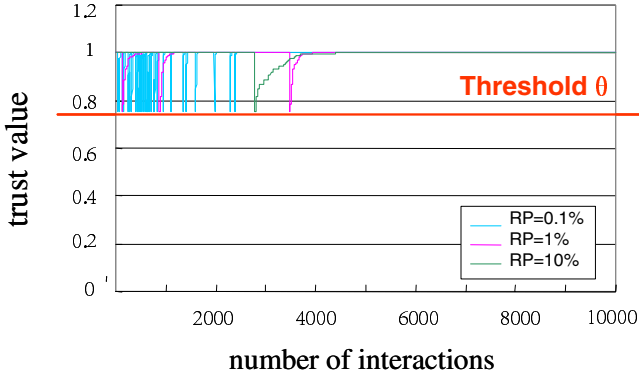


Fig. 5. This shows a figure simulated trust value for various RP peer rates

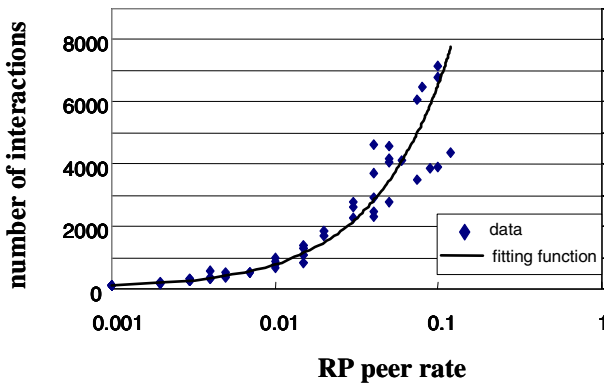


Fig. 6. Simulation result for number of interactions versus RP peer rate

Figure 5 shows how many interactions were required when the illegal peer rate was set to 3%. It also compares the RP rate, which ranged from 0.1% to 10%. When RPs found the illegal peers, the trust value fell according to Eq. (1).

Note that for the line for an RP rate of 0.1%, the interval for detecting illegal peers becomes wider. Although it needed about 5500 interactions to detect all illegal peers when the RP rate was set to 0.1%, the number of interactions required was not very different when it was under 1%. We also found that the results were scattered when the RP rate was less than 1%, but there was not much difference when it was over 1%.

Figure 6 shows how many interactions were required until the RPs’ trust value could be restored when the illegal peer rate was set to 3%. It took more time to restore the trust value to 1 when the RP rate was higher. This is because an RP waited for other RPs’ transactions. This suggests that the number of RPs is more than required, so there are many peers that do not have sufficient tasks.

Figure 7 shows how many interactions were required when the RP rate was set 1% and the illegal peer rate ranged from 0.1% to 25%. RPs could detect efficiently when there were few illegal peers, and they could do it in about 3000-4000 interactions no matter how much the number of illegal peers increased.

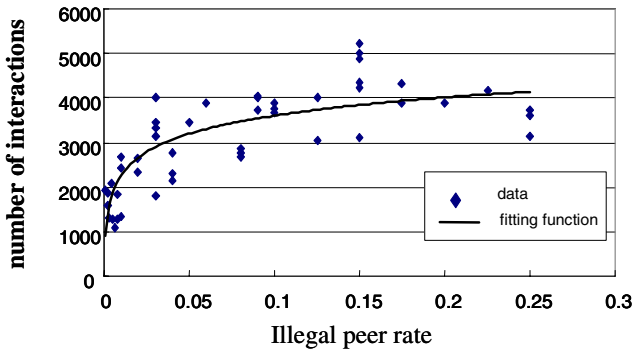


Fig. 7. Simulation result for number of interactions versus illegal peer rate

#### 4.2 Analyses and Consideration

The simulation results show that the proposed algorithm could augment the trust value of each RP by evaluating behavior in the community. From our results, the most efficient number of RPs was 1% of the total number of peers. Also, RPs could detect all illegal actions among 3000-4000 interactions even if there were many illegal peers. If the rate of illegal peers in the community is less than 1%, then 3000 interactions are needed. This suggests that 1% of RPs out of all participants is the most effective number.

Moreover, this algorithm can be applied to ordinary peers. Because the load on RPs increases as the community size grows, the ordinary peer layer should be divided into several layered groups of RPs, as shown in Figure 8. In each group, selected ordinary peers act as RPs, and if these RPs cannot perform trust evaluation, they ask the upper RPs for evaluation. This autonomous cooperation of peers has advantages of localization and network-wide load balancing as well as load reduction in the RP layer, as discussed above.

This algorithm identifies peers that are falsifying metadata or content. However, community actions after the detection depend on the community’s policy. For example, illegal peers could be removed from the community or communication could be blocked at RPs or at proxies that can be controlled by the community operator. At least, warnings could be sent to the illegal peers.

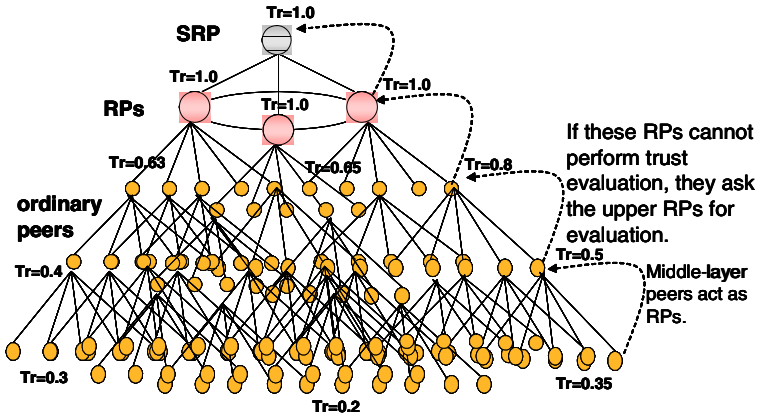


Fig. 8. This shows N-layered autonomous trust tree

### 4.3 Service Applications

The proposed illegality suppression mechanism can be applied to service operations in the future ubiquitous content exchange communities.

- Content ranking

Because cIDs are collected in RP peers, the content ranking can be created based on the cID data. The ranking data can be used as incentives to creators to create more attractive content. This service is expected to contribute to the enlargement of ubiquitous content exchange markets.

- User information management

Through data-mining of collected cID data, preferences can be identified. The results can be used to recommend content or services. Also, more specific communities can be established for people who have close preferences or properties.

## 5 Conclusion

This paper has addressed a new method for establishing a more credible ubiquitous P2P content exchange community. Although rigid digital rights management in a P2P community has already been proposed, it is effective mainly for preventing illegal behavior of end users, for example, content falsification or illegal copying. Here, we proposed a more efficient method for processing content distribution. It aims to control content exchange so that illegal activities can be reduced to a sufficiently low level. By observing content exchange interactions in rendezvous points, this method can determine whether illegal activities have been performed and which peers did them. Simulation results showed the most efficient number of RPs was 1% of the total number of peers.

We also showed some promising applications that use the data collected through this mechanism. We expect our proposals to contribute to the creation of the ubiquitous communication society.

## Acknowledgement

This work was supported in part by the Research and Development Program of Ubiquitous Network Authentication and Agent (2004), the Ministry of Internal Affairs, and Communications (MIC), Japan.

## References

1. MIC, Japan, "white paper on telecommunications 2005", <http://www.johotsusintokei.soumu.go.jp/whitepaper/eng/WP2004/2004-index.html>
2. T. Iwata, et al, "A DRM System Suitable for P2P Content Delivery and the Study on its Implementation," APCC 2003.
3. Content ID forum <http://www.cidf.org/>
4. Creative Commons <http://creativecommons.org/>
5. Digital Commons <http://digitalcommons.jp/>
6. L. Lessig, "The future of ideas," Random House, New York, 2001.
7. N. S. Good and A. Krekelberg, "Usability and privacy: A study of Kazaa P2P file-sharing," June 2002. Available at <http://www.hpl.hp.com/shl/papers/kazaa/index.html>
8. O. Avaro and P. Salembier, "MPEG-7 Systems: Overview," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 11, No. 6, June 2001.
9. T. Oh-ishi, K. Sakai, T. Iwata, and A. Kurokawa, "The Deployment of Cache Servers in P2P Networks for Improved Performance in Content-Delivery," IEEE P2P 2003.
10. Y. Wang, J. Vasseileva, "Trust and Reputation Model in Peer-to-Peer Networks," IEEE P2P2003.
11. L. Xiong and L. Liu, "A Reputation-Based Trust Model for Peer-to-Peer eCommerce Communities," IEEE P2P2003.
12. R. Chen and W. Yeager, "Poblano: Distributed Trust Model for Peer-to-Peer Networks," <http://www.jxta.org/docs/trust.pdf>
13. B. F. Cooper, M. Bawa, N. Daswani, and H. Garcia-Molina, "Protecting the PIPE from malicious peers," Stanford Technical Report, May 2002.

# Location-Based Routing Protocol for Energy Efficiency in Wireless Sensor Networks

Hyuntae Cho and Yunju Baek

Department of Computer Science and Engineering, Pusan National University,  
Busan, Republic of Korea

marine@juno.cs.pusan.ac.kr, yunju@pusan.ac.kr

**Abstract.** Energy efficiency in wireless sensor networks is an important design consideration. We present a location-based routing protocol for energy efficiency in wireless sensor networks called GPSR-S (Greedy Perimeter Stateless Routing for wireless Sensor networks). GPSR-S is based on GPSR, which is one of the most well-known location-based routing protocols for wireless ad hoc networks. We improve the energy efficiency of GPSR by considering nodes' energy level and location information. In addition, we modify the address-centric nature of the algorithm into a data-centric one. Simulation results show that GPSR-S performs well in terms of energy efficiency and the number of packets. GPSR-S delivers approximately 10% fewer packets than GPSR, but the lifetime of the network is 10% greater.

## 1 Introduction

Wireless sensor networks are likely to be widely deployed in the future because they greatly extend the ability to monitor and control the physical environment from remote locations. Such networks are composed of a large number of sensor nodes, which are densely deployed either inside the phenomenon or very close to it. These tiny sensor nodes, which consist of sensing, data processing, and communicating components, use the idea of sensor networks based on the collaborative effort of a large number of nodes. Some of the application areas are health, military, and security.

The position of sensor nodes need not be engineered or pre-determined. This means that sensor network protocols and algorithms must possess self-organizing capabilities. So, wireless sensor networks usually use the protocols and algorithms for wireless ad hoc networks in order to achieve self-organization. Location awareness is another important issue in wireless sensor networks. Most data collection is based on location, so it is desirable that the nodes know their position whenever it is needed. In most cases, location information is needed in order to calculate the distance between two particular nodes so that energy consumption can be estimated. Since there is no addressing scheme (such as IP-addresses) for sensor networks and they are spatially deployed on a region, location information can be utilized in routing data in an energy efficient way.

Although GPSR[3] has been proposed for traditional wireless ad hoc networks, it is not well suited to the unique features and application requirements of sensor

networks. To illustrate this point, the differences between sensor networks and ad hoc networks are outlined below:

- The number of sensor nodes in a sensor network can be several orders of magnitude higher than the nodes in an ad hoc network.
- Sensor nodes are more densely deployed.
- Sensor nodes are limited in power, computational capacity, and memory.
- Sensor nodes may not have global identification because of the large overhead and large number of sensors.
- Sensor networks are “data-centric” i.e., unlike traditional networks where data is requested from a specific node, data is requested based on particular attributes, such as, “which area has temperature  $> 50^{\circ}\text{F}$ ?” Data-centric protocols are query-based and depend on the naming of desired data, which helps in eliminating many redundant transmissions.

For these reasons, GPSR should be fixed before it is applied in wireless sensor networks. So, we propose a modified version of GPSR called GPSR-S (GPSR for wireless Sensor networks) to fulfill the requirements of wireless sensor networks. GPSR-S uses energy-aware and geographical information to route a packet towards the target region. Within a region, it uses a forwarding technique to disseminate the packets. Next, we modify wireless sensor networks to be data-centric, where communications are expressed, not in terms of node identifier but in terms of named data.

The remainder of the paper is organized as follows: In Section 2, we discuss related work on wireless sensor networks. In Section 3, we explain location-based routing for energy efficiency in wireless sensor networks. In Section 4, we simulate GPSR-S and compare its performance to GPSR. Section 6 concludes.

## 2 Related Work

Routing protocols in wireless sensor networks are classified into various categories: (i) address-centric and data-centric routing, with respect to method of data acquisition; (ii) flat and hierarchical routing, with respect to the form of the network; and (iii) location-based and location-independent routing protocols.

In data-centric routing, the sink sends queries to certain regions and waits for data from the sensors located in the selected regions. Since data is requested through queries, attribute-based naming is necessary to specify the properties of data. SPIN[6] is the first data-centric protocol, which considers data negotiation between nodes in order to eliminate redundant data and save energy. Later, Directed Diffusion[5] was developed and constituted a breakthrough in data-centric routing. Directed Diffusion is an important milestone in data-centric routing research on sensor networks.

In common with other communication networks, scalability is a major design attribute of sensor networks. A single-tier network can cause the gateway to overload with an increase in sensor density. So, single-gateway architecture is not scalable for a larger set of sensors covering a wider area of interest, because the sensors are typically not capable of long-haul communication. The main aim of hierarchical routing is to efficiently maintain the energy consumption of sensor nodes by involving them in multi-hop communication within a particular cluster and by the sink. Cluster forma-

tion is typically based on the energy reserve of sensors and sensor's proximity to the cluster head. LEACH(Low-Energy Adaptive Clustering Hierarchy)[7] is one of the first hierarchical routing approaches for sensor networks. The idea proposed in LEACH has been an inspiration for many hierarchical routing protocols such as TEEN(Threshold sensitive Energy Efficient sensor Network protocols)[7], and APTEEN(Adaptive Periodic TEEN)[8].

Most of the running protocols for sensor networks require location information for sensor nodes. Since there is no addressing scheme for sensor networks and they are deployed spatially across a region, location information can be utilized in routing data in an energy-efficient way. For instance, if the region to be sensed is known, using the location of sensors, the query can be diffused only to that particular region, which will reduce the number of transmissions significantly. GPSR[3] is one of the most well known geographic routing algorithms in wireless ad hoc networks. Other hierarchical schemes in use are GEAR (Geographical and Energy Aware Routing)[9], and LAR (Location Aided Routing)[10].

GPSR uses the position of nodes and a packet's destination to make decisions about packet forwarding. GPSR makes greedy forwarding decisions, using only information about a node's immediate neighbors in the network topology. When a packet reaches a region where greedy forwarding is impossible, the algorithm recovers by routing around the perimeter of the region. By keeping state only about the local topology, GPSR scales better in per-node state than shortest-path and ad-hoc routing protocols as the number of network destinations increases. Under mobility's frequent topology changes, GPSR can use local topology information to find correct new routes quickly. But, when GPSR is used in wireless sensor networks, it has some shortcomings. Because it uses greedy forwarding to distance from the node to the destination, the specific nodes which located on the optimum path can only consume energy. Also, it reduces energy efficiency due to address-centric routing. In order to fulfill the above requirements, in the next section we propose a location-based routing protocol for energy efficiency.

### **3 Location-Based Routing Protocol for Energy Efficiency in Wireless Sensor Networks**

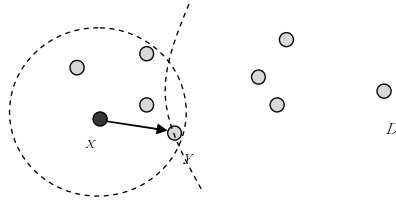
Wireless sensor networks are composed of a number of nodes that have limited energy resources. The energy depletion of nodes in wireless sensor networks brings about the partitioning of the network. To avoid such partitioning, the consumption of energy in wireless networks must be distributed fairly across all nodes. Furthermore, the main aim at the network layer is to find ways for energy-efficient route setup and reliable relaying of data from the sensor nodes to the sink so that the lifetime of the network is maximized. So, the forwarding process that we propose consists of two phases: (1) forwarding node selection phase, and (2) data dissemination inside the target region.

#### **3.1 Forwarding Node Selection Technique**

Packets are marked by their originator with their destinations' locations. Upon receiving a packet, a node checks its neighbors to see if there is one that is closer to



the destination. If such a neighbor exists, GPSR-S picks a next-hop node among all neighbors that are closer to the destination than itself. Here, each node knows its own location and remaining energy level, and its neighbors' location and remaining energy levels through a simple "neighbor hello" protocol. To select a closer neighbor to the destination, GPSR-S uses information about nodes' energy level and location.



**Fig. 1.** Forwarding node selection

A formula proposed for the next forwarding node selection is as follows:

$$next\ hop = \max \left( \alpha \frac{dist(N, D) - dist(Ni, D)}{r} + (1 - \alpha) \frac{Er(Ni)}{Ef(Ni)} \right) \text{ [formula 1]}$$

Where  $Ni$  is the neighbor node of node  $N$ ,  $D$  indicates the centroid of the target region, and  $dist(x, y)$  is the distance between node  $x$  and node  $y$ .  $r$  is the initial radio transmission coverage,  $Er(Ni)$  is the remaining energy level of node  $Ni$ , and  $Ef(Ni)$  is the initial energy level of all nodes. In formula 1,  $\alpha$  can be adjusted to emphasize either the minimizing path length to the destination or the balancing energy consumption.  $\alpha$  has a value between 0 and 1. Note that each node can obtain its location information, energy level, and its neighbor's information, which presumably are already available due to the needs of sensor network applications through hello messages. After selecting a forwarding node, the forwarding node selection process repeats until the packet reaches the target region.

The power of this forwarding to route using only neighbor nodes' positions come with one attendant drawback: there are topologies in which the only route to a destination requires that a packet move temporarily farther in geometric distance from the destination. We address the failure of pure greedy routing to find paths in the presence of *voids*, by introducing perimeter traversal algorithms (used in GPSR) for forwarding packets around *voids*.

After describing the right-hand rule for traversing a graph, GPSR characterizes the behavior of the right-hand rule on wireless network graphs, and observes the role played by crossing edges in the graph in interfering with the right-hand rule traversal. GPSR then introduces perimeter probing while employing a no-crossing heuristic to eliminate crossing edges from the graph, and uses the resulting state to forward around *voids*.

### 3.2 Data Dissemination Within Destination Region

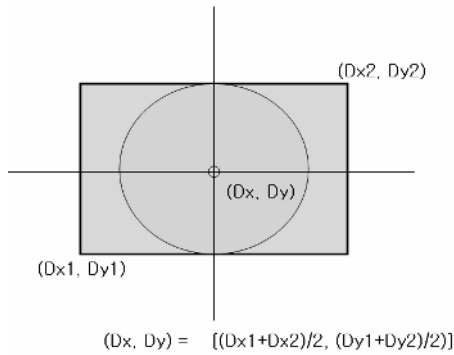
As mentioned above, the wireless sensor networks use data-centric routing instead of address-centric routing. GPSR sends queries to one node instead of multiple nodes. For the description of the algorithm, we assume a rectangular region specification. The form of query sent by sink is below.

```

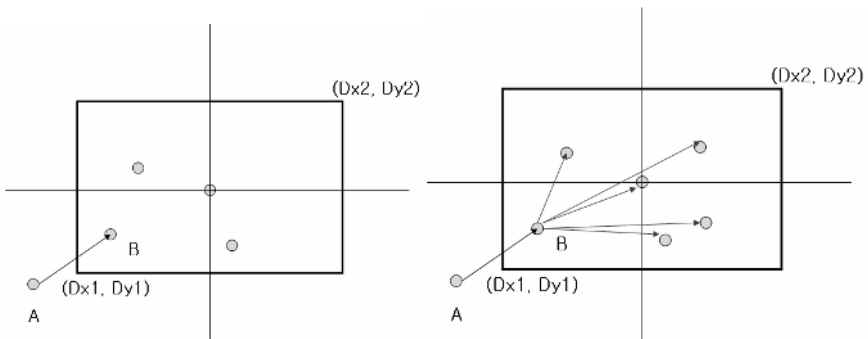
type = four-legged animal
rect = [100, 200, 200, 400]
etc...
    
```

Each query packet has a target region specified in some way. The centroid of the target region can be calculated as follows:

$$\text{Centroid: } (Dx, Dy) = [(Dx1 + Dx2) / 2, (Dy1 + Dy2) / 2] \quad [\text{formula 2}]$$



**Fig. 2.** The centroid of the destination region

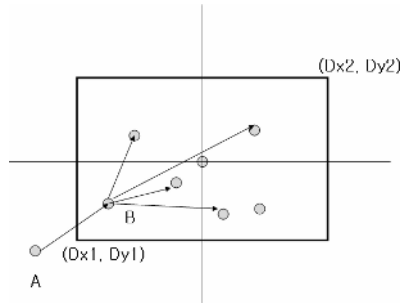


**Fig. 3.** Data dissemination inside the target region

As shown in Figure 3, after the first packet reaches the destination, B receives a packet from A, and finds itself inside the target region. If the current node is the only one inside this region, the packet is forwarded within this region. Once the packet is

inside the target region, a simple flooding with duplicate suppression scheme can be used to flood the packet inside the target region.

However, flooding is expensive in terms of energy consumption, due to the fact that in this simple flooding scheme, every node has to broadcast once and all its neighbors receive this broadcast message. This is especially expensive in high-density networks. Therefore, we can use a RGF(Recursive Geographic Forwarding) approach to disseminate the packet inside the target region[9].



**Fig. 4.** Recursive Geographic Forwarding

RGF supposes that the target region is the big rectangle, and now node receives a packet for a region, and finds itself inside this region. In this case, node B creates four new copies of a packet bound to 4 subregions(as shown by the four small rectangles in figure 4). This recursive splitting and forwarding procedure is repeated until the stop condition is satisfied.

## 4 Simulation Results

We study and compare GPSR with GPSR-S. In addition, we evaluate how our algorithm compares with GPSR. Since the original GPSR does not handle routing to a region, it does not consider energy efficiency. We have augmented GPSR to route packets to a region, thus considering energy efficiency. The performance measure is increased network lifetime due to routing in a target region and energy awareness.

To evaluate performance, we used NS-2[12]. The test model is MICA2 mote, designed by UC Berkeley. It operates at 433MHz radio frequency, and consumes from 5.3mA to 26.7mA in Tx mode and 7.4mA in Rx mode. The antenna is 10cm high, it is omni-directional, and its transmission range is 10m.

In the study reported in this paper, we varied network size, density and other parameters. More specifically, the simulation results shown in this section include networks ranging in size from 30 to 150. For network simulation, its geometric area was  $50 \times 50$  square. The initial location of each node was randomly generated. The 802.11 was used in the radio model. The initial data size was 42bytes and the transmission period was between 0.5 and 2sec. The simulations were run for 100

seconds, the speed of nodes was set to 1m/sec, and the initial energy of each node was 10J.

Figure 5 shows the number of live nodes up to 20. This graph considers only energy efficiency. That is, it uses only a forwarding node selection method. This result shows that when 100 seconds had elapsed, 52 nodes in GPSR, 72 nodes in GPSR-S( $\alpha=0.3$ ), and 75 nodes in GPSR-S( $\alpha=0.4$ ) were alive. Intuitively, this means that GPSR-S has a longer network lifetime than GPSR. The reason that GPSR-S prolongs network lifetime compared with GPSR, is that GPSR-S tends to concentrate on greedy forwarding to distance and energy efficiency.

Figure 6 shows the remaining energy distribution of nodes when 2/3 of the entire lifetime has elapsed. Herein, the number of nodes is 90. As shown in the graph, it is evident that the nodes in GPSR-S consume the energy more fairly than those in GPSR.

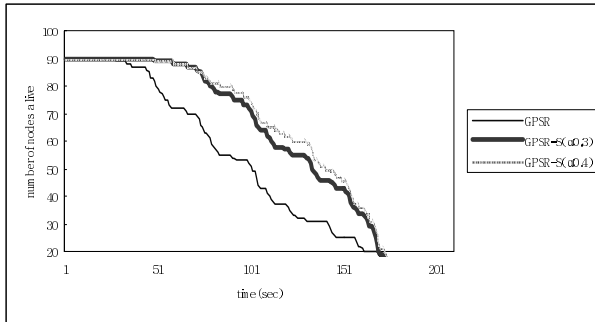


Fig. 5. The number of live nodes

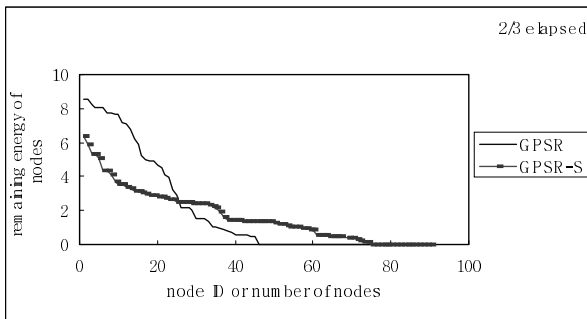


Fig. 6. Energy distribution of nodes within the network

Figure 7 shows the number of packets generated according to  $\alpha$ . From the graph, it can easily be seen that the number of packets varies according to the value of  $\alpha$  and the form of network. Also, a higher  $\alpha$  can generate more traffic within the network.

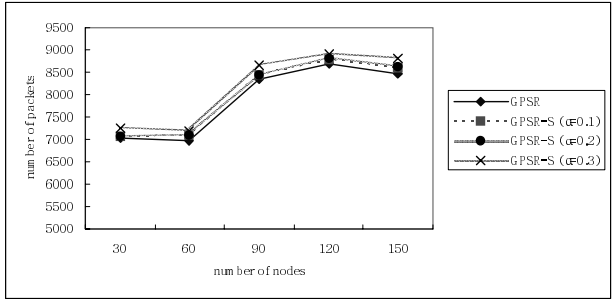


Fig. 7. The number of packets to values of  $\alpha$

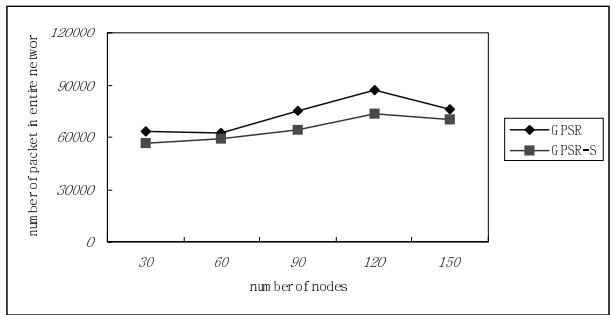


Fig. 8. The number of packets in the entire network

Figure 8 shows the number of packets that are generated in the entire network by applying data-centric routing, which has a target region. There are three nodes in this region. In this case, G-PSR-S has 10% less traffic than G-PSR.

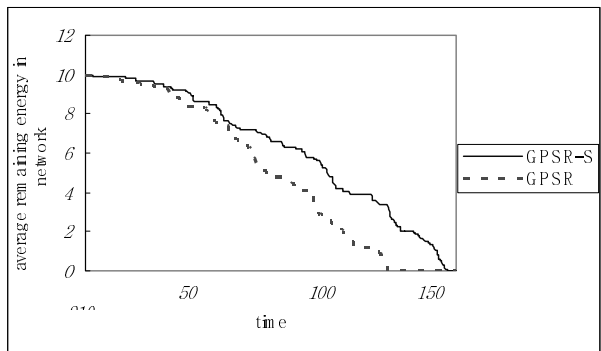


Fig. 9. The average remaining energy of all nodes

Figure 9 compares GPSR with GPSR-S with respect to energy consumption over time. The performance of GPSR-S is higher with respect to energy consumption and longevity of the network. After 160sec, 1% of all nodes in GPSR and 3% of all nodes in GPSR-S remain. It is clear that the amount of energy consumed increases according to the number of packets.

Because our algorithm relies on random decisions, it is important to show that its performance does not vary significantly over several runs.

## 5 Conclusion and Future Work

The traditional routing protocols are not considered to be energy-efficient, so only specific nodes on the optimum path consume energy. We studied the problem of forwarding a packet to nodes in a geographic region of a wireless sensor network. The proposed GPSR-S protocol uses energy-aware and geographic neighbor selection to route a packet towards the target region. Within a region, it uses a forwarding technique to disseminate the packet. These strategies attempt to balance energy consumption and thereby increase network lifetime.

Simulation results show that the GPSR-S consumed energy uniformly while 20% of the totality of nodes were alive. Data centric routing, moreover, use less traffic than the routing mechanisms which communicate with a specific node. GPSR generates traffic per node as it sends data to nodes. In GPSR-S, about 10% can be reduced in traffic by adapting data-centric routing. However, there is a possibility of delay in data transmission. GPSR-S, which is proposed in this paper, results in a better protocol for wireless sensor networks because it reflects the energy efficiency and data-centric routing appropriately. This strategy attempts to balance energy consumption and thereby increase network lifetime. We evaluated the performance through simulation. In further work, we intend to investigate how real implementation affects the performance of the protocol.

**Acknowledgement.** This work was supported by the Regional Research Centers Program(Research Center for Logistics Information Technology), granted by the Korean Ministry of Education & Human Resources Development.

## References

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, pp. 393-422, 2002.
- [2] Kemal Akkaya, Mohamed Younis, "A survey on routing protocols for wireless sensor networks," *Ad Hoc Networks*, pp. 26, 2003.
- [3] B. Karp and H.T. Kung. "GPSR: Greedy perimeter stateless routing for wireless networks," In *ACM MOBICOM*, Boston, MA, August 2000.
- [4] W. Rabiner Heintzelman, A. Chandrakasan, and Hari Balakrishnan. "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," In *Proceedings of the 33rd International Conference on System Sciences (HICSS '00)*, 2000.

- [5] Chalermek Intanagonwiwat, Ramesh Govindan, Deborah Estrin, John Heidemann, Fabio Silva, "Directed diffusion for wireless sensor networking," *IEEE/ACM Transactions on Networking (TON)*, Volume 11, Issue 1, pp : 2 - 16, 2003.
- [6] Joanna Kulik, Wendi Rabiner Heinzelman, Hari Balakrishnan, "Negotiation-based Protocols for Disseminating Information in Wireless Sensor Networks," *ACM Wireless Networks*, 1999
- [7] Arati Manjeshwar et al., "TEEN: A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks," *Proc. Second Int'l Workshop Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*, 2001.
- [8] Arati Manjeshwar et al., "APTEEN: A Hybrid Protocol for Efficient Routing and Comprehensive Information Retrieval in Wireless Sensor Networks," *IEEE Proc. of the International Parallel and Distributed Processing Symposium(IPDPS'02)*, Apr. 2002, pp.195-202.
- [9] Yan Yu, Ramesh Govindan, Deborah Estrin "GEAR : Geographical and Energy Aware Routing : a recursive data dissemination protocol for wireless sensor networks," *Technical Report UCLA/CSD-TR -01-0023*, UCLA Computer Science Dept., May 2001.
- [10] Y.-B. Ko and N. H. Vaidya. "Location-aided routing (LAR) in mobile ad hoc networks," In *ACM MOBICOM*, October 1998.
- [11] <http://www.icir.org/bkarp/GPSR/GPSR.html>
- [12] Network Simulator 2 : <http://www.isi.edu/nsnam/ns/>
- [13] <http://www.zigbee.org>

# Efficient Access of Remote Resources in Embedded Networked Computer Systems

Paul S. Usher and Neil C. Audsley

Real-Time Systems Research Group,  
Department of Computer Science, University of York, York YO10 5DD, UK  
usher@cs.york.ac.uk

**Abstract.** Fixed networks of limited resource heterogeneous computers need to allow applications to access remote devices in order to overcome any local resource deficiencies. Current operating systems would use a file system, network stack and middleware to implement such access but the volume of functionality involved can be a barrier to performance. This paper examines the 2.4 series Linux kernel to show that networked operating systems lack flexibility and performance in this environment. It also presents a low level approach that can reduce the overheads incurred and improve performance when remote devices are accessed.

## 1 Introduction

This paper is concerned with fixed networks of heterogeneous single processor embedded and ubiquitous computing devices operating in close proximity and in networks that are largely static in nature. This means that only one type of communication medium may ever be used and the physical address of a resource may often be known in advance. The primary issue is therefore how to structure the operating system (OS) so that it is able to quickly and efficiently facilitate access to any remote devices required by the application in order to overcome local inadequacies. Such problems can in specific circumstances be addressed using additional hardware, both SMP and NUMA architectures are evidence of this approach but such designs are not in the context of this paper.

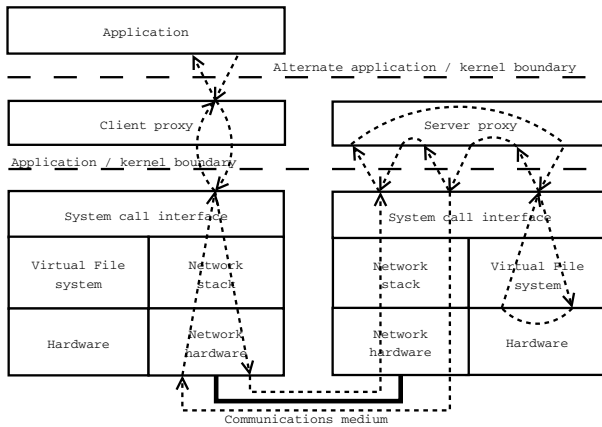
This paper analyses the 2.4 series of Linux kernel to illustrate the performance of a typical networked OS. It then shows how a low level approach that embeds a simple file based protocol directly into an Ethernet frame can make better use of the hardware characteristics improving both performance and flexibility.

The remainder of this paper is structured as follows; section 2 examines the architecture of the Linux kernel in order to show why a networked OS is not ideally suited to this tightly constrained environment. Section 3 supports this with a performance analysis of a typical GNU Linux based OS. Finally section 4 outlines how the large packet sizes of IEEE 802.3 and IEEE 802.11 can be combined with the PSE51 embedded systems profile of IEEE 1003.13 to remove much of the complexity involved with accessing remote devices [1, 2, 3].



## 2 Background: A Networked OS

A Networked OS such as Linux uses a stack of software for process control and communication, another for resource access (VFS) and another for remote communication (network stack) [4, 5]. Layers of functionality are then used within these stacks to aid flexibility and further simplify implementation. Accessing a remote device using this model requires the client machine to have some functionality (a proxy) connecting the file system and the network stack so that the application can access a remote device in the same manner as a local device. Alternatively if a separate interface is used the functionality must implement the operations via the existing network stack. The server also requires similar functionality in order to access the device on behalf of the remote application. Consequently a considerable amount of functionality is involved as the flow of control passes up and down the file system and network stacks on both the client and server machines, see Figure 1.



**Fig. 1.** Control flow when accessing remote resource access

To address the performance issues of this model the proxy processes can be moved into the address space of the kernel in order to reduce the amount of copying and the number of context switches, but this does not reduce the volume of code involved in navigating the VFS or network stack [4, 5]. The network stack also does not allow the characteristics of the delivery device to affect the operation of the layers above it [4, 5]. Consequently local socket based communication is likely to be adversely affected as the fragmentation and redelivery functionality cannot be removed when the delivery mechanism is reliable (memory). It therefore seems clear that networked OSs are not ideally suited to constrained environments where flexible and efficient access is required to both local and remote resources (devices, files or applications).

### 3 Performance Analysis of a Linux OS

To demonstrate the performance of a typical networked OS this section uses the Lmbench suite of benchmarks (version 2.04) on an isolated and directly connected 10Mb two node network of otherwise idle Slackware Linux based computers [6]. The specification of the test machines is outlined in Table 1.

**Table 1.** Test machine specification

	Darkstar	Cheetah
CPU	AMD K6-2 350MHz	Intel Pentium II-MMX 266MHz
Memory	128MB	64MB
Hard disk	WDC AC28400R	WDC AC34000L
Network Card	3Com 3c590 10 BaseT	3Com 3c905C-TX/TX-M
OS	Slackware 9.0.0	Slackware 10.1.0
Kernel Version	2.4.22	2.4.29

The bandwidth of a network medium will govern the performance and characteristics of any networked system but its importance escalates as more advanced functionality is added. The addition of remote device access capabilities to a networked system begins to make it more like a distributed system, as a result the performance of the network medium becomes critical to the successful operation of the system as more and more applications become increasingly addicted to remote resources.

The availability of necessary quantities of network bandwidth clearly governs how much work a system can get done, however, it is also important not to achieve this at the expense of latency. Both figures will ultimately be dominated by the performance characteristics of the network medium and associated hardware. Although the sheer volume of functionality involved in the file system, network stack and proxy components of the OS is likely to have a limiting affect on the performance of an application, even if this only occurs at high levels of load.

#### 3.1 Bandwidth Results

The bandwidth benchmark transmits 3MB's of data between two processes in 48KB steps and then returns it once the data has been received. This is performed primarily between processes on the same machine since the majority of inter process communication is local, although processes on different machines are used if a particular mechanism supports remote communication, see Table 2.

A Pipe is a very simple communication mechanism supporting one way local communication and it is not surprising that it is the best performer. In contrast a UNIX socket provides two way local communication and achieves only 60-75% of a Pipes throughput. TCP sockets differ from their UNIX counterparts by

**Table 2.** Communication bandwidth (MB/s)

Communications Mechanism	Local		Remote	
	Darkstar	Cheetah	Darkstar	Cheetah
Pipe	82.8	143.0	-	-
UNIX Sockets	61.9	85.4	-	-
TCP Sockets	43.8	68.8	1.05	1.03

supporting remote communication across heterogeneous and unreliable communications mediums and when they are used in this fashion it is not surprising that they perform poorly in comparison to either UNIX sockets or pipes. However, they perform equally poorly for local communication (50% of Pipe performance) as the upper layers of the IPv4 stack remain unchanged even though the majority of their functionality is not required in the local environment. The effect of this performance loss should not be underestimated as it may not be possible to choose in advance an alternate communication mechanism in order to statically optimize performance. It might therefore be beneficial if the kernel was able to to optimise performance wherever possible, possibly by passing a UNIX socket of as a TCP socket.

### 3.2 Latency Results

The latency benchmark uses a 1 byte “hot-potato” token that allows the resulting TCP or UDP message to fit into the minimum Ethernet frame (46 bytes of payload). It also tests Sun’s RPC mechanism when used with both TCP and UDP and this makes it possible to better estimate the overheads incurred by a proxy process in a networked or distributed environment. Such functionality acts as the glue that binds the network stack and file system models together and an estimate of its performance therefore gives a more realistic view of an applications performance when remote devices are accessed, see Table 3.

Distributed systems require a suitable balance between bandwidth and latency. It is therefore concerning that TCP sockets incur 4-6 times the latency of a pipe when used in a local environment, whilst UNIX and UDP sockets incur 2-4 times the latency in the same situation.

**Table 3.** Communications latencies ( $\mu$ s)

Communications Mechanism	Local		Remote	
	Darkstar	Cheetah	Darkstar	Cheetah
Pipe	24.4	13.7	-	-
UNIX Sockets	52.2	25.6	-	-
UDP Sockets	77.6	55.2	231.7	235.5
TCP Sockets	107.8	88.0	279.1	280.4
Sun RPC over UDP	180.7	150.7	325.3	328.6
Sun RPC over TCP	230.7	204.8	425.9	434.3
TCP connection	416.0	340.0	451.3	476.3

**Table 4.** RPC overhead ( $\mu$ s)

Communications mechanism	Local		Remote	
	Darkstar	Cheetah	Darkstar	Cheetah
Sun RPC over UDP	103.1	95.5	93.6	93.1
Sun RPC over TCP	122.9	116.8	144.6	154.3

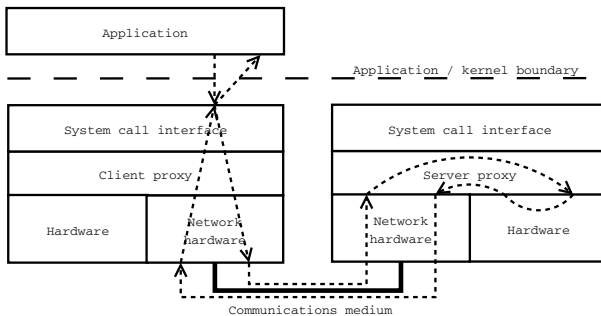
When considering the additional overheads involved in accessing remote devices it is important not to forget that a TCP connection must be established prior to its use and these results suggest that this takes around 340-470 $\mu$ s. This coupled with the costs of an RPC call over the same mechanism would seem to rule out the use of TCP for client/server style connections in any networked environment where the connection is not established for a considerable period of time.

The overheads for the RPC mechanism are relatively constant regardless of both the communication mechanism (UDP or TCP) and the location of the client (local or remote). For UDP based communication this overhead is around 95 $\mu$ s, whilst TCP communication sees this increase slightly to 120-150 $\mu$ s, See Table 4.

## 4 Reducing Overheads

This section examines whether an OS whose architecture directly targets the need to access remote devices might reduce overheads, improve performance and still achieve sufficient flexibility.

Communications mediums such as IEEE 802.3 and 802.11 allow computers to deliver well over 1KB of data in an error free manner because of the capacity of the packet and the use of a 32bit CRC [1, 2]. If the majority of interactions could be made to fit into a single packet additional reliable delivery functionality may never be needed. In addition the embedded systems profile of IEEE 1003.13 (PSE51 of POSIX.13) indicates that a traditional file system is unnecessary



**Fig. 2.** Control flow when accessing remote resource access

in such systems. Instead sufficient flexibility is achieved by interfacing devices directly to the `close()`, `open()`, `read()` and `write()` functions, thus negating the need for the majority of functionality associated with a file system [3]. It therefore seems worthwhile to examine whether the relatively large packet sizes supported by these mediums can be utilised to directly encapsulate sufficient information to allow one OS to send file system requests directly to another OS without the use of either a VFS or the network stack. This reduction in the systems footprint would also allow it to be utilised in more restricted environments in addition to reducing the latency of any remote device access. The architecture of the resulting system is illustrated in Figure 2.

#### 4.1 Format of an Ethernet Based File Protocol

The following fragments of C code outline the format of the Ethernet frames when the parameters for the file operations are embedded directly into the data payload. This starts by outlining the format of an Ethernet header, and the types of messages supported by the protocol.

```
#include<stdint.h>

/* Declare an Ethernet header */
struct ethhdr {
    uint8_t dst[6];
    uint8_t src[6];
    uint16_t type;
};
/* Declare a file protocol instruction */
struct fp_inst {
    struct ethhdr header;
    uint16_t opcode; /* Instruction type */
    uint32_t tag; /* Instruction number */
};
/* Types of instruction used in the file protocol */
enum fp_opcode {
    FP_CLOSE,
    FP_OPEN,
    FP_READ,
    FP_REPLY,
    FP_WRITE
};
```

The “opcode” field uniquely identifies the type of instruction contained in the Ethernet frame in addition to aligning all of the following data onto a 32bit boundary in order to maximise performance. The client allocates the following “tag” field in order to uniquely identify the request and to allow it to match a reply to the appropriate request. This kind of approach is used with some success in both 9P and Styx as it allows the server to identify incoming requests that

have been repeatedly made by the client in order to overcome the unreliability of a networked system. The following code illustrates how the parameters for the various functions are encoded.

```

/* Declare a close instruction */
struct fp_close {
    struct fp_inst type;
    uint32_t fd;
};
/* Declare an open instruction */
struct fp_open {
    struct fp_inst type;
    uint32_t flags;
    uint32_t mode;
    uint8_t filename [];
};
/* Declare a read instruction */
struct fp_read {
    struct fp_inst type;
    uint32_t fd;
    uint32_t len;
};
/* Declare a write instruction */
struct fp_write {
    struct fp_inst type;
    uint32_t fd;
    uint32_t datalen;
    uint8_t data [];
};

```

The successful operation of a close(), open(), read() or write() function call may result in some data being returned to the caller and potentially some error code. In addition to this the read() needs to return some data to the client. The following structure could be used to represent this information.

```

/* Declare a reply instruction */
struct fp_reply {
    struct fp_inst type;
    uint32_t result; /* Return value */
    uint32_t error; /* Error code */
    uint32_t datalen; /* Length of returned data */
    uint8_t data []; /* Returned data */
};

```

## 4.2 Payload Utilisation

The file protocol outlined here provides a mechanism for efficiently allowing an application to access a device connected to a remote computer. Since the

computers are all on the same network and the file protocol does not support messages bigger than a single frame it is possible to dramatically reduce the size of the headers required, see Table 5.

**Table 5.** Per layer comparison of protocol overheads in bytes

OSI Layer	Protocol		
	UDP	TCP	FP
Network	20	20	6
Transport	8	20	0
Total	28	40	6

The supported file operations require little data so they all fit comfortably into even the smallest Ethernet packet. This is particularly beneficial since it allows the maximum amount of data to be carried by those messages that also support a dynamic data portion. An open request for example supplies a file name, the length of which is only known at run time. Similarly it is not possible to know how much data will be written to a file, or how much may be returned from a read operation. Since all of this data must be contained in a file protocol message and these cannot span multiple Ethernet frames it is important that the dynamic portion of these messages is as large as possible so as not to reduce flexibility, see Table 6.

**Table 6.** Maximum size of dynamic data portion in bytes

Ethernet payload	Message		
	open	reply	write
46	32	28	32
1500	1486	1482	1486

The file name passed to `open()` can therefore be between 31 and 1485 bytes in length, since the last byte must be null in order to terminate the string. Whilst a single Ethernet frame is able to incorporate in excess of 1400 bytes for both the read and write operations. Transferring more data than this would necessitate breaking the larger operation up into multiple smaller requests, in addition to some support from the protocol for atomic actions so that either the whole request succeeds or it fails.

### 4.3 Operation of the Server Proxy

The results obtained from the testing of the Linux kernel made it quite clear that there are significant overheads involved with the packaging and un-packaging of data prior to it being sent to the server. The approach adopted here reduces these by limiting the amount of additional data that the protocol needs as well

as only supporting a very small set of operations. This minimalist approach allows the operation of the server to be simplified. Handling of each type of incoming message (reply is outgoing) is offloaded to a function that is dedicated to the purpose. Dispatching is then a simple matter of checking that the opcode field is valid before using it as an index into an array of function pointers. Of the operations undertaken by the server the most expensive is read since it requires the allocation of sufficient memory to hold the data that is read from the file prior to it being sent back to the client.

The initial implementation of the protocol communicates with the remote machine via a packet socket that has been bound to a specific network connection (typically eth0). This allows both low level access to raw Ethernet packets and simplifies the implementation process, as well as allowing its performance to be analysed on the same two node Slackware Linux based system as has been used for the UDP/TCP analysis. Further development work is underway to integrate this functionality at a lower level within the kernel in order to further reduce any overheads. As a result the performance of both the client and particularly the server is likely to improve in the future.

#### 4.4 Performance Analysis of Initial Implementation

The performance of this initial implementation has been measured through the use a benchmarking application that runs on one of the Slackware Linux machines whilst accessing files on the other via the file protocol implementation. The latencies are therefore those typically experienced by the application, see Table 7.

**Table 7.** Remote file operation performance

	Latency ( $\mu$ s)
close()	236
open()	242
read()	1535
write()	1695

The use of minimum sized Ethernet frames in both directions ensures that close() and open() perform better than read() and write(), which (in this case) utilise a full packet in one direction in order to maximise the amount of data transferred (see Table 6). The parameters supplied to these operations can have a significant affect on their performance, open() in particular requires an additional 440 $\mu$ s when the truncate flag is used.

Given that this is an initial implementation these performance figures compare quite favourable with the 325-440 $\mu$ s required to transmit a single byte of data via Sun's RPC mechanism on UDP or TCP, especially as this figure does not account for any overheads incurred when the server process accesses the local files on behalf of the remote application.



## 5 Conclusions

It has been shown that access to remote devices requires some form of proxy and that current designs incur significant overhead both in terms of establishing a reliable connection, and in marshaling the data. It therefore seems unlikely that such software designs will achieve acceptable performance in embedded systems without the support of additional hardware resources. Although small network stacks are undoubtedly available it is dubious whether they provide any practical benefit in this context since the primary barrier seems to be performance rather than size. In addition it has been shown that a traditional networked OS provides unnecessary functionality in some areas and insufficient support in others. It has also been shown that there may be a potential size and performance benefit if the architecture of the OS makes better use of the resources it has available to it. To demonstrate this fact we have provided a simple Ethernet based file protocol that facilitates efficient access to remote devices with sufficient flexibility to satisfy the file based functionality required of a POSIX.13 PSE51 compliant system.

## References

1. IEEE: IEEE 802.3-2002: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) access method and physical layer specifications. (2002)
2. IEEE: IEEE 802.11-1999: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. (1999)
3. IEEE: 1003.13-1998 IEEE Standard for Information Technology — Standardized Application Environment Profile (AEP) — POSIX® Realtime Application Support. (1998)
4. Bovet, D.P., Cesati, M.: Understanding the Linux Kernel. Second edn. O'Reilly & Associates, Inc. (2002)
5. Rubini, A., Corbet, J.: Linux Device Drivers. Second edn. O'Reilly & Associates, Inc. (2001)
6. McVoy, L.W., Staelin, C.: lmbench: Portable tools for performance analysis. In: USENIX Annual Technical Conference. (1996) 279–294

# Lightweight Ontology-Driven Representations in Pervasive Computing

Jarosław Domaszewicz and Michał Rój\*

Institute of Telecommunications, Warsaw University of Technology,  
Nowowiejska 15/19, 00-665 Warsaw, Poland  
j.domaszewicz@tele.pw.edu.pl, m.roj@tele.pw.edu.pl

**Abstract.** A clearly specified representation of diverse entities is needed to refer to them in pervasive computing applications. Examples of such entities include physical objects, operations, sensor and actuator resources, or logical locations. We propose a novel way to systematically generate representations of entities for programmable pervasive computing platforms made of tiny embedded nodes. Our original idea is to generate a very lightweight, though semantically-rich, representation from a possibly complex ontological specification. At the platform development phase, a domain ontology is used to describe the target environment. A preprocessing tool produces the ontology-driven, lightweight representation, which comes in two flavors: a human-readable one, to be used for programming, and a binary one, to be used at runtime. Our approach makes it possible to take advantage of all the benefits of ontology-based modeling and, at the same time, to obtain a representation light enough to be embedded in even the tiniest nodes.

## 1 Introduction

Pervasive computing applications may need to refer to a great variety of entities to perform their tasks<sup>1</sup>. Categories of such entities include physical objects, operations to be performed on those objects, various resources (e.g., sensors and actuators), and logical locations. The applications refer to entities to discover, control, or use them in some way. By “representation” we mean any agreed upon convention enabling the applications to unambiguously refer to entities. A programmable pervasive computing platform should clearly specify how to represent entities that might be of interest to the applications. Application programmers need a human-friendly representation (e.g., descriptive identifiers or function names) to refer to entities in their source code. At runtime, deployed applications and the pervasive computing platform need a common binary representation.

In many pervasive computing platforms the representation of entities is not generated in a systematic way. A common practice is to produce a representation in an ad-hoc manner, e.g., by arbitrary assignment of identifiers to arbitrarily selected entities.

---

\* The order of authors was determined by a coin flip.

<sup>1</sup> The research reported in this paper has been partly supported by the Polish Ministry of Scientific Research and Information Technology, grant no. 3T11D 011 28.

We believe that a good representation of entities in pervasive computing should result from a systematic procedure. Specifically, the representation should be derived from an explicit, formal domain model of the pervasive computing platform's target environment (i.e., the domain where the applications run). The model should be comprehensive enough to capture all relevant aspects of the domain, including possible diversity of entities. To ensure high quality of the model, it should be created by domain experts, not programmers.

Producing a so-called ontology is an excellent way to model a domain. Ontologies are in widespread use in Semantic Web, and they have been successfully used in pervasive computing. However, most ontology-based techniques require relatively big amounts of memory and processing power, especially when a domain ontology itself is used at runtime (which is the case for all ontology-based pervasive computing systems known to us).

Applying ontologies becomes challenging if the target pervasive computing platform consists exclusively of tiny, energy-constrained, battery-powered nodes, like Berkeley Motes [1]. As in such a system there is no room for any big central repository or server, the representation must be stored and processed locally, by the nodes themselves. It has to be extremely lightweight and compact. This case is the focus of the paper.

This paper makes the following contribution. We propose to methodically generate ontology-driven representations of entities for pervasive computing platforms made of tiny embedded nodes. Our original idea is to produce a very lightweight, though semantically rich, representation of entities (down to binary encodings) from a possibly complex ontological specification.

The paper is organized as follows. In Section 2, we discuss related work. In Section 3 we present our approach in general terms. Section 4 gives an example of a domain ontology and a lightweight ontology-driven representation and how the final representation is acquired. In Section 5, we give a specific example of how it can be used to represent sensor and actuator resources in a pervasive computing middleware. The paper is concluded in Section 6.

## 2 Related Work

Ontology-based domain modeling has an established position in the field of pervasive computing. For example, a so-called GAS ontology is used to model the functionality of devices [2]. The ontology, which aims at augmented home objects (such as "eLamp," or "eBook"), defines operations that can be performed on devices (e.g., switch on/off). In a different approach [3], capabilities (sampling rate, physical units, etc.) of sensor nodes in a wireless sensor network are ontologically described, and the ontology is used mainly to dynamically calibrate the whole system.

In the above examples, the ontology itself is present at runtime. Handling an ontology directly is definitely not suitable for small, Berkeley Mote-class nodes. Even if the ontology is kept as small as possible (e.g., GAS-CO in the GAS architecture [2]), and lightweight ontology languages are employed (e.g., OWL-Lite in [3]), a node has to be more a PDA than a mote. The primary difference between our approach and the above ones is that in our case not the ontology itself but an ontology-derived lightweight representation suitable for tiny embedded nodes is used at runtime.

We also identified several areas of ontology-derived software artifacts - in fields other than pervasive computing. Usually, ontological models are used to generate a class hierarchy in object-oriented languages (especially Java). For example, Jena API [4] includes a program called *schemagen*<sup>2</sup>, which generates Java representations of concepts from an OWL [5] ontology. Also, a class generator has been included into the Protégé OWL plugin<sup>3</sup>. A general approach to mapping OWL into Java is discussed in [6]. Algorithms and simple heuristics to generate class diagrams from ontologies are presented in [7]. An ontology editor able to generate knowledge from ontologies in various formats (including Java classes) is introduced in [8].

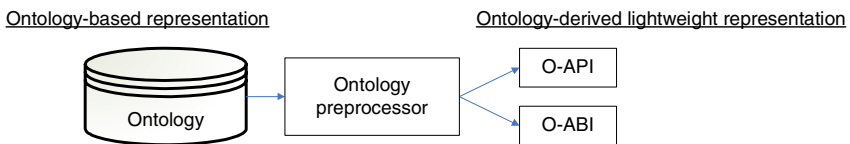
Even though our approach shares some similarities with the above ones, representations of entities generated with our approach are lighter and more elementary and so can be aimed at tiny embedded nodes. In particular, we claim that exploring the possibility of deriving simple (yet semantically meaningful) binary encodings from a complex, abstract ontology has not been done before.

A very recent, promising attempt to employ ontologies in software engineering is presented in [9]. Our work can contribute to those efforts.

The techniques presented in this paper can be applied in existing pervasive computing platforms, in which the representations are produced in an ad-hoc way. To the best of our knowledge such platforms include Agilla [10], tinyDB [11], tinyLIME [12], as well as many others.

### 3 Lightweight Ontology-Driven Representations

Our key idea is presented in Fig. 1. We propose a tool, called “ontology preprocessor,” that takes a domain ontology as its input and produces a lightweight representation for a category of entities as its output. The representation comes in two parts, named O-API and O-ABI.



**Fig. 1.** Deriving lightweight representation from an ontology

The ontology is created by a domain expert. It describes the target environment (e.g., home, office), where a pervasive computing platform is to be deployed. In particular, it may classify and describe different kinds of objects, logical locations, or resources (e.g., sensors and actuators) that are common in the target environment. The ontology can be as big and complex as desired.

The ontology preprocessor, which derives the lightweight representation, is not tied to any specific ontology. It is ontology-independent, so ontologies for various

<sup>2</sup> <http://jena.sourceforge.net/how-to/schemagen.html>

<sup>3</sup> <http://protege.stanford.edu/plugins/owl/>

domains can be used as its input. We are currently developing algorithms for ontology preprocessing. Some preliminary results are presented in the next section.

The ontology preprocessor produces the representation of entities in two flavors: O-API and O-ABI. They bear some resemblance to, but are not the same as, a regular API and ABI (i.e., Application Programming Interface and Application Binary Interface, respectively). The main difference between O-API and O-ABI is that the former is targeted at humans (programmers) and the latter at machines (compilers and the runtime system).

O-API is meant to be used by programmers to refer to entities in source code. In O-API, entities are represented by human-readable, meaningful names, e.g., constant identifiers. O-API should be simple enough to be usable by a programmer without any background in ontology engineering. Even though full O-API can be big (its size grows with that of the ontology), any single application is likely to use only a small subset.

O-ABI is meant to be used by applications to refer to entities when interacting with the system software of the pervasive computing platform. In O-ABI, entities are represented by simple binary encodings “understood” by the system software. Normally, there is a one-to-one correspondence between O-API names and O-ABI encodings.

The O-ABI representation is lightweight in that it can be embedded into even most severely constrained nodes. While full O-ABI can be quite big (just as the ontology and O-API), any single node is likely to be related to only a fairly limited number of entities. For example, an intelligent node embedded into an object is usually equipped with a handful of sensors and actuators. Handling a limited number of binary encodings is possible even if a node’s processing power, memory, or available energy are extremely scarce.

## 4 Deriving O-API and O-ABI from Ontology: An Example

We now provide a simple example of what the input ontology and the derived lightweight representation may look like. Consider a pervasive computing platform where the entities to be represented are operations that can be performed on home objects by embedded nodes. A part of a home domain ontology for this case might be the one presented in Fig. 2. The ontology consists of three basic hierarchies: *HomeItem*, *Location*, and *Operation*. They classify home objects (for brevity we include light sources and meters only), logical locations of the objects, and possible operations, respectively.

All the *Operation* instances either affect an object’s state (the *ControlOperation* operations) or to inquire about it (the *ObserveOperation* operations). The discrete state variables are handled with *SwitchOperation* and *ObserveSwitchOperation* operations, while the continuous state variables with *TurnOperation* and *ObserveKnobOperation* operations. The operations are linked to home item classes using the *controls* and *observes* properties. For example, since *ControlOperation* is linked to *HomeItem* using the *controls* property, *LightSource* can be controlled with any *ControlOperation* (note that properties are “inherited” here). In our ontology we assumed that only permanently attached objects (the *FixedLightSource* class) have a logical location.

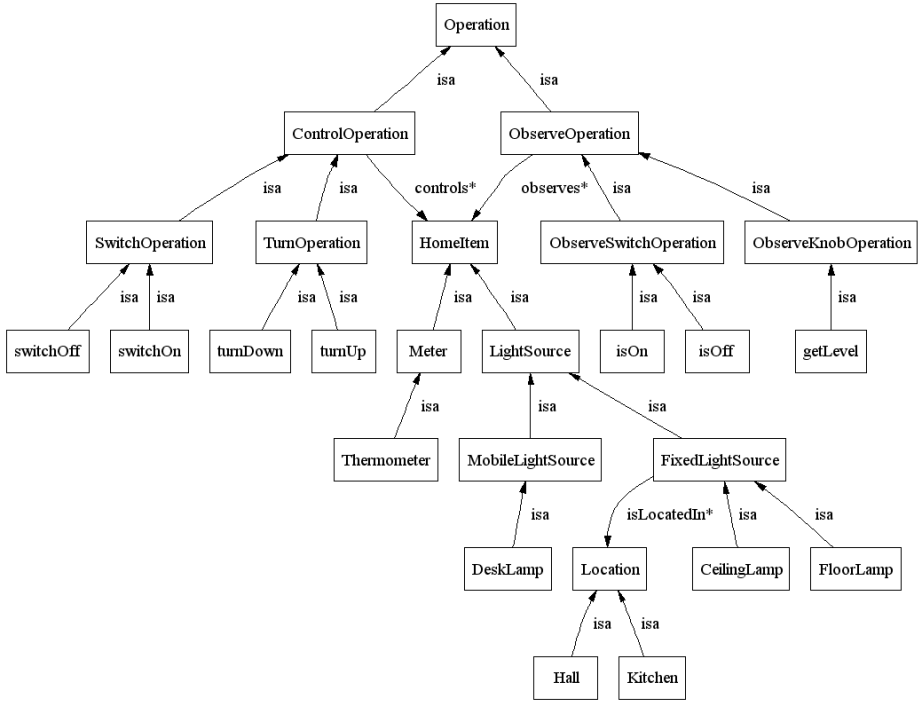


Fig. 2. A simple home domain ontology

A part of a lightweight representation that could be produced from this ontology by an ontology preprocessor is shown in Tab. 1. To produce entities we used a simple algorithm. For every “concrete” operation class (turnUp, turnDown, switchOn, switchOff, isOn, isOff and getLevel) we follow the controls and observes properties. We explore all possible paths from an Operation class to a HomeItem class (example paths are turnUp-controls-FloorLamp and getLevel-observes-Thermometer). Then we go further, by following the isLocatedIn property. For instance, for fixed light sources we explore paths consisting of three classes, such as turnUp-controls-FloorLamp-islocatedIn-Hall and turnUp-controls-FloorLamp-isLocatedIn-Kitchen. Then we can assign an API name to every path (e.g., turnUp-controls-FloorLamp-islocatedIn-Hall becomes turnUpFloorLampInHall). For a more complex ontology, the algorithm can follow longer paths and produce more semantically rich operations.

Proper ontological modeling ensures that all the paths (and corresponding entities) are meaningful. For example, for mobile light sources there are no paths that include a logical location and so an entity like switchOnMobileLightSourceInKitchen is not produced by the algorithm. As another example, assume the controls operation is restricted not to take on values in the Meter class (the restriction not shown in Fig. 2). Then entities like turnUpThermometer are not produced.

**Table 1.** O-API and O-ABI pairs for the category of operations, based on the ontology presented in Fig. 2. O-ABI encodings have been selected arbitrarily.

O-API	O-ABI
switchOnHomeItem	0x00
...	...
switchOnLightSource	0x10
...	...
switchOnMobileLightSource	0x40
...	...
switchOnFixedLightSource	0x50
...	...
switchOnFixedLightSourceInHall	0x60
switchOnFixedLightSourceInKitchen	0x61
...	...
switchOnCeilingLamp	0x70
...	...
switchOnCeilingLampInHall	0x80
...	...

Some remarks are in order at this point. The entities (in this case – operations) are no longer selected in an arbitrary fashion. The set of entities is systematically derived from the ontology. Thus, ontology preprocessing produces not only representations of entities, but, in a sense, the entities themselves.

In our example, there is no direct mapping between entities and existing classes of the domain ontology (or their instances). Rather, the entities are “produced” by manipulating and combining concepts present in the domain ontology. Thus, the ontology processor can be considered as a value adding tool. We are working on an approach allowing the entities to be expressed in OWL, as new concepts based on the ones present in the original ontology.

Different relationships captured by the ontology may be reflected in some structuring of the derived set of entities. One example is a hierarchical structuring. For example, the increasing specificity in the object hierarchy leads to increasingly specific operations (compare `switchOnLightSource`, `switchOnFixedLightSource`, and `switchOnCeilingLamp`). Whenever an object has an additional attribute (e.g., a logical location), an even more specific operation (e.g., `switchOnCeilingLampInKitchen`) can be produced. Another example of structuring the set of entities is semantically organizing their binary encodings (explained in Section 5).

The entities might include quite complex and abstract concepts (“semantic richness”). For example, `switchOnCeilingLampInKitchen` conveys information on what activity is to be taken (switching on), what the object of the operation is (a ceiling lamp), and where the operation is to be performed (in the kitchen). Even such semantically rich entities are ultimately represented by simple O-ABI encodings.

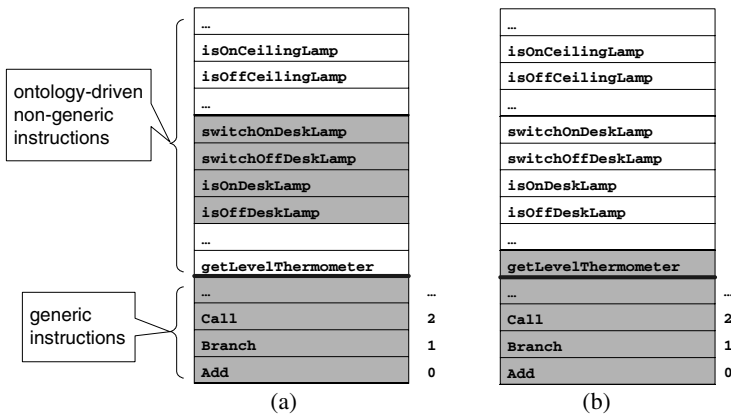
Even though the example covers the category of operations, our approach could be applied to other categories of entities as well. Some obvious examples are the objects themselves (e.g., `KitchenCeilingLamp`), sensors and actuators embedded in the objects (e.g., `DeskLampSwitch`), logical locations (e.g., `HallWithWallLamp`), and events generated by users (e.g., `CeilingLampSwitchedOn`). The choice of category depends on the architecture of a pervasive computing platform.

A lightweight representation of entities, like the one presented in Fig. 3., can in principle be produced by hand in an ad-hoc manner. However, such a task quickly becomes unfeasible and the resulting representation hardly maintainable, even if the domain is described with only a moderate number of concepts.

### 5 ROVERS: Exposing a Node’s Resources Through a Virtual Machine

In this section we provide an example of applying a lightweight, ontology-driven representation in a pervasive computing platform. The example is based on ROVERS – a middleware that we are developing [13]. The middleware targets peer-to-peer networks of constrained heterogeneous nodes. The nodes are embedded in everyday objects and equipped with different combinations of resources, primarily sensors and actuators.

In ROVERS, applications are composed of tiny collaborating mobile code units, called micro-agents<sup>4</sup>. To enable code mobility, a virtual machine is deployed on each node. As nodes differ in terms of resources they are equipped with, so do their virtual machines. A node’s virtual machine is specified by instructions it can execute. All possible instructions are classified into generic and non-generic. Generic ones, like arithmetic operations or program flow instructions are supported by the virtual machine on every node.



**Fig. 3.** An example of the instruction encoding space for ROVERS virtual machines. A specific virtual machine supports the shadowed items only: (a) a desk lamp, (b) a thermometer.

In ROVERS, a node’s sensor and actuator resources are represented by (non-generic) instructions of the node’s virtual machine. For example, a temperature sensor may be represented by the `getLevelThermometer` instruction, while a desk lamp’s switch by the `switchOnDeskLamp`, `switchOffDeskLamp`, `isOnDeskLamp`, and

<sup>4</sup> We cover only those aspects of ROVERS that are directly related to the topic of this paper.



`isOffDeskLamp` instructions. A node's virtual machine supports only those non-generic instructions that represent the node's sensors and actuators.

An instruction (generic or non-generic) has a human-readable name and a binary encoding. The names constitute a specific assembly language, while the encodings – a machine language. What is unique is that the non-generic parts of the both languages are derived from a domain ontology as O-API and O-ABI, respectively.

An instruction encoding space may look like the one presented in Fig. 3. The instructions shadowed in Fig. 3 (a) and (b) might be supported by nodes embedded into a desk lamp and a thermometer, respectively. As each node is equipped with only a small number of sensors and actuators, handling the ontology-derived representation of resources amounts to interpreting a couple of encodings.

### 5.1 Semantically-Structured Binary Encodings

The instruction encoding space presented in Fig. 3 does not suggest any structuring of the binary encodings (O-ABI). However, deriving the encodings from an ontology, as advocated in this paper, gives rise to an additional benefit – being able to automatically organize them based on their semantics. This subsection gives an example of such O-ABI structuring. The goal of this particular one is to reduce the size of ROVERS micro-agents' binaries (and so the energy cost of their mobility).

The structuring is based on the following observation. Consider the home environment domain. Assume that the ontology classifies domain concepts into sub-domains. Examples might include the “physical” sub-domain (temperature, pressure, humidity, etc.), the lighting sub-domain (ceiling lamps, desk lamps, etc.), or the heating sub-domain. Since each micro-agent should do a single job well, non-generic instructions used by most micro-agents are likely to originate exclusively from a single sub-domain. For example, a temperature reporting micro-agent might use the `getLevelThermometer` instruction, originating from the “physical” sub-domain, as its only non-generic instruction. Similarly, a “light manager” micro-agent would use instructions from the lighting sub-domain. Of course, a complete application will likely include micro-agents working in different sub-domains.

The above observation could be used to structure the encoding space of the ROVERS virtual machine instructions. Assume there are no more than 128 generic instructions and no more than 128 non-generic ones originating from a single sub-domain. Then the instruction encoding space could be the one presented in Fig. 4.

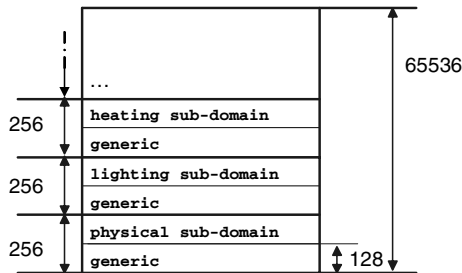


Fig. 4. Semantically-structured O-ABI

The space is divided into 256-instruction segments. The lower half of each segment is occupied by the generic instructions (which in effect have multiple encodings), while the upper half belongs to non-generic ones originating from a single sub-domain. Assume there is a generic “segment prefix” instruction that specifies the segment that the subsequent instructions belong to (until the next segment prefix instruction). Then, in spite of the fact that the encoding allows as many as  $2^{16}/2 = 32768$  non-generic instructions, each instruction can be encoded with only 8 bits. Obviously, the average encoding efficiency (number of bits per instruction) depends on how often the segment prefix instruction is used. Since a typical micro-agent is limited to a single sub-domain, only one segment prefix instruction per micro-agent is needed, and the average encoding efficiency approaches 8 bits per instruction.

Even though the example pertains to the ROVERS middleware, semantic structuring of binary encodings seems to be a promising technique of wider applicability. It is easy to implement as a feature of O-ABI generation.

## 6 Summary and Future Work

This paper presents a novel approach to using ontologies in the development of pervasive computing platforms. We made initial experiments with generating representations from OWL ontologies, using the Jena API [4]. Our plan now is to develop more advanced ontology preprocessing algorithms. One of the challenges is to make them ontology-independent, so ontologies for various domains can be used as inputs. In addition, we plan to make the ontology preprocessor tunable so that, for example, the “granularity” of generated entities can be specified as a parameter.

## References

1. Hill, J., Horton, M., Kling, R., and Krishnamurthy, L.: The Platforms Enabling Wireless Sensor Networks. *Communications of the ACM*, 2004. 47(6).
2. Christopoulou, E., Goumopoulos, C., Zaharakis, I., and Kameas, A.: An Ontology-based Conceptual Model for Composing Context-Aware Applications. in *Workshop on Advanced Context Modelling, Reasoning and Management in conjunction with Sixth International Conference on Ubiquitous Computing (UbiComp 2004)*. 2004. Nottingham, England.
3. Avancha, S., Patel, C., and Joshi, A.: Ontology-driven Adaptive Sensor Networks. in *First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous '04)*. 2004. Boston, USA.
4. McBride, B.: Jena: A Semantic Web Toolkit. *IEEE Internet Computing*, 2002. 6(6): p. 55 - 59.
5. Bechhofer, S., et al.: OWL Web Ontology Language Reference, <http://www.w3.org/TR/owl-ref/>, July 2005
6. Kalyanpur, A., Pastor, D.J., Battle, S., and Padget, J.: Automatic mapping of OWL ontologies into Java. in *Sixteenth International Conference on Software Engineering and Knowledge Engineering (SEKE)*. 2004.
7. Bonancin, R. and Baranauskas, C.C.: From Ontology Charts to Class Diagrams: Semantic Analysis Aiding Systems Design. in *International Conference on Enterprise Information Systems*. 2004. Porto, Portugal.

8. Mian, P.G. and de Almeida Falbo, R.: Building Ontologies in a Domain Oriented Software Engineering Environment. in IX Congreso Argentino de Ciencias de la Computación. 2003. La Plata, Argentina.
9. Tetlow, P., et al.: Ontology Driven Architectures and Potential Uses of the Semantic Web in Software Engineering, <http://www.w3.org/2001/sw/BestPractices/SE/ODA/>, June 2005
10. Fok, C.-L., Roman, G.-C., and Lu, C.: Rapid Development and Flexible Deployment of Adaptive Wireless Sensor Network Applications. in 24th International Conference on Distributed Computing Systems (ICDCS'05). 2005. Columbus, Ohio, USA.
11. Madden, S., Franklin, M.J., Hellerstein, J.M., and Hong, W.: The design of an acquisitional query processor for sensor networks. in International Conference on Management of Data. 2003. San Diego, California.
12. Curino, C., et al.: Tiny Lime: Bridging Mobile and Sensor Networks through Middleware. in 3rd IEEE International Conference on Pervasive Computing and Communications (PerCom 2005). 2005. Kauai Island, Hawaii.
13. ROVERS Homepage, <http://meag.tele.pw.edu.pl/ROVERS/index.htm>, July 2005

# Object Tracking Using Durative Events

Eiko Yoneki and Jean Bacon

University of Cambridge Computer Laboratory,  
Cambridge CB3 0FD, United Kingdom  
{Eiko.Yoneki, Jean.Bacon}@cl.cam.ac.uk

**Abstract.** This paper presents a distributed middleware architecture based on a service-oriented approach, to manage high volume sensor events. Event management takes a multi-step operation from event sources to final subscribers, combining information collected by wireless devices into higher-level information or knowledge. An event correlation service provides sophisticated event filtering, aggregation and correlation over time and space in heterogeneous network environments. An experimental prototype in the simulation environment with real world data produced by the Active BAT system is shown.

## 1 Introduction and Background

The majority of current middleware for Wireless Sensor Networks (WSNs) is based on the data centric approach, and a fundamental idea naturally came from database management systems (DBMS). The database community has taken the view that declarative programming, through a query language, provides the right level of abstraction for accessing, filtering, and processing relational data. The middlewares that take a database approach such as [3] provides an interface for data collection but they do not provide general purpose distributed computation. For example, it is complex to implement arbitrary aggregation and filtering operators and communication patterns with query languages. Thus, more general interfaces for global network programming are desirable.

Recent evolution of ubiquitous computing with a dramatic increase of event monitoring capabilities by wireless devices and sensors requires an open platform for users to utilize seamlessly various resources in physically interacting environments, unlike the traditional closed network setting for specific applications. There has been an effort to architect middleware for such environments using service oriented architecture (e.g. RUNES [6] and P2PComp [1]). When designing the middleware for sensor networks, heterogeneity of information over global distributed systems must be considered. The sensed information by the devices is aggregated and combined into higher-level information or knowledge.

Service Oriented Architecture (SOA) is a well proven concept for distributed computing environments. It decomposes applications, data, and middleware into reusable services that can be flexibly combined in a loosely coupled manner. SOA maintains agents that act as software services performing well-defined operations. This paradigm enables the users to be concerned only with the operational

description of the service. All services have a network addressable interface and communication via standard protocols and data formats (i.e., messages). SOA can deal with aspects of heterogeneity, mobility and adaptation, and offers seamless integration of wired and wireless environments.

Generic service elements are context model, trust and privacy, mobile data management, configuration, service discovery, event notification, and the following are the key issues addressed for our design.

- Flexible discovery mechanisms for ad hoc networks, which provide the reliable discovery of newly or sporadically available services.
- Support for adaptive communication modes, which provides an abstract communication model underlying different transport protocols. Notably, event-based communication is suitable for asynchronous communication.

Peer-to-peer networks and grids offer promising paradigms for developing efficient distributed systems and applications. Grids are essentially P2P systems. The grid community recently initiated a development effort to align grid technologies with Web Services: the Open Grid Services Architecture (OGSA) [4] lets developers integrate services and resources across distributed, heterogeneous, dynamic environments and communities. The OGSA model adopts the Web Services Description Language (WSDL) to define the concept of a grid service using principles and technologies from both the grid and Web Services. The architecture defines standard mechanisms for creating, naming, and discovering persistent and transient grid-service instances. The convergence of P2P and Grid computing is a natural outcome of the recent evolution of distributed systems, because many of the challenging standards issues are quite closely related.

The Open Services Gateway Initiative (OSGi) [5] is focused on the application layer and open to almost any protocol, transport or device layers. The three key aspects of the OSGi mission are multiple services, wide area networks, and local networks and devices. Key benefits of the OSGi are that it is platform independent and application independent. In other words, the OSGi specifies an open, independent technology, which can link diverse devices in the local home network. The central component of the OSGi effort is the services gateway. The services gateway enables, consolidates, and manages voice, data, Internet, and multimedia communications to and from the home, office and other locations.

We propose a distributed middleware architecture that envisages an integrated service oriented architecture to manage high volume sensor events in global computing. The current mainstream deployment of sensor networks collects all the data from the sensor networks and stores them in the database and data analysis is preceded from there. The proposed architecture deploys distributed gateways to collaborate data management over hybrid network environments. The publish/subscribe paradigm is used for asynchronous communication, performing data aggregation and distributing filtered data to other networks based on contents. We simulate distributed gateways with the real world data produced by the Active BAT system [2].

This paper continues as follows: section 2 describes the middleware architecture, section 3 briefly recalls the durative event model defined in our previous

work [7], section 4 reports an experimental prototype, section 5 describes related works and it concludes with section 6.

## 2 Middleware Architecture

We have developed a generic reference architecture applicable to any ubiquitous computing space. The middleware contains separate physical, sensor components, event broker, service, and service management layers including an open application interface. An implementation of a reference architecture is in progress.

A service is an interesting concept to be applied in WSNs. It may be a role on a sensor node, or a function providing location information. Services allow cascading without previous knowledge of each other, and enable the solution of complex tasks, where functional blocks are separated in order to increase flexibility and enhance scalability of sensor network node functions. A key issue is to separate the software from the underlying hardware and to divide the software into functional blocks with a proper size and functionality. Another important issue is that the sensed data should be filtered, correlated, and managed at the right time and place when they flow over heterogeneous network environments. It is not easy to provide reliable and useful data among the massive information from WSNs. An event correlation based on our previous work [7] is integrated with service composition.

### 2.1 Service Semantics

Service semantics is an important issue, in addition to the service definition, so that services can be coordinated in the space. The model of the real world is of a collection of objects, where objects maintain state using sensor data, and applications' queries and subscriptions are a relevant sets of objects. Fig.1 shows an example of object mappings among applications, middleware and sensor components. Objects are tightly linked to event types in an event broker. Exploiting semantics will let the pervasive space's functionality and behaviour develop and evolve. Space specific ontologies will enable such exploitation of knowledge and semantics in ubiquitous computing.

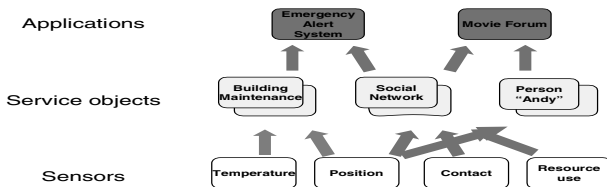


Fig. 1. Mapping Real World to Applications

### 2.2 Layer Functionality

Fig.2 depicts the overview of the middleware, and the brief functionality of each layer is shown below.

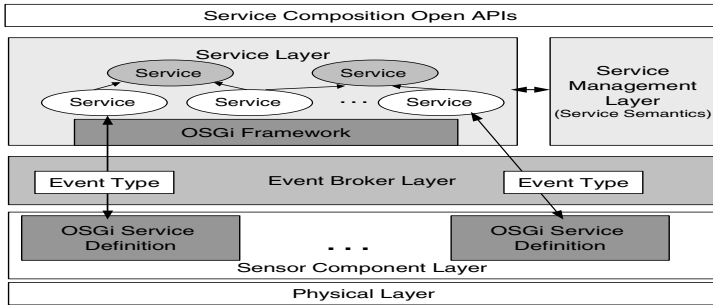


Fig. 2. Middleware Architecture with Wireless Sensor Data

**Physical Layer:** This layer consists of the various sensors and actuators.

**Sensor Component Layer:** A sensor component layer can communicate with a wide variety of devices, sensors, actuators, and gateways and represent them to the rest of the middleware in a uniform way. A sensor component effectively converts any sensor or actuator in the physical layer to a software service that can be programmed or composed into other services. Decoupling sensors and actuators from sensor platforms ensures openness and makes it possible to introduce new technology as it becomes available.

**Event Broker Layer:** This layer is a communication layer between Sensor components and the Service layer. It supports asynchronous communication using the publish/subscribe paradigm. Event filtering, aggregation, and the correlation service is a part of this layer.

**Service Layer:** This layer contains the Open Services Gateway Initiative (OSGi) framework, which maintains leases of activated services. Basic services represent the physical world through sensor platforms, which store service bundle definitions for any sensor or actuator represented in the OSGi framework. A sensor component registers itself with the service layer by sending its OSGi service definition. Application developers create composite services via the Service Management Layer's functions to search existing services and using other services to compose new OSGi services. Canned services, which may be useful globally, could create a standard library.

**Service Management Layer:** This layer contains an ontology of the various services offered, and the appliances and devices connected to the system. Service advertisement and discovery use service definitions and semantics to register or discover a service. Service definitions are tightly related to the event types used for communication in the Event Broker Layer including composite formats. The reasoning engine determines whether certain composite services are available.

**Application Interface:** An application interface provides open interfaces for applications to manage services, including the management of contexts.

### 3 Durative Events and Interval-Based Semantics

In [7], we defined a unified semantics, combining traditional event composition and data aggregation in WSNs. For event detection, we introduced a parameterized algebra. Parameters include time, selection, consumption, precision, and subset policies. This approach defines unambiguous semantics of event detection and supports resource constrained environments. The semantics is integrated with the service composition engine in the middleware described in Section 2.

In our event model, a primitive event is the occurrence of a state transition at a certain point in time. Each event has a timestamp associated with the occurrence time. The timestamp is an approximation of the event occurrence time. In most event algebras, each event occurrence, including composite events, is associated with a single value indicating the occurrence time. This may result in unintended semantics for some operator combinations, for example nested sequence operators. We define a composite event with duration and give a new interval-based timestamp to a composite event based on an interval semantics. An interval-semantics supports more sensitive interval relations among events in environments where real-time concerns are more critical, such as wireless networks or multi-media systems. An event can have a space stamp indicating certain location, relative location, and grouping (e.g. position (x,y,z), global id). Complex timing constraints among correlated event instances are precisely defined (see Appendix in [7]).

Composite events are defined by expressions built from primitive and composite events and algebraic operators. Operators consist of *Conjunction*, *Disjunction*, *Concatenation*, *Sequence*, *Concurrency*, *Iteration*, *Negation*, *Selection*, *Spatial Restriction*, and *Temporal Restriction* (See [7] for the detail). We also support parameters, which help to define unambiguous semantics of event detection and support resource constrained environments. In resource constrained network environments, the event algebra must be restricted so that only a subset of all possible occurrences of complex events will be detected, and this can be achieved by applying appropriate parameters. The following example illustrates the use of the operators to describe composite events.

**Example:** The temperature of rooms with windows facing south is measured every minute and transmitted to a computer placed on the corridor.  $T$  denotes a temperature event and  $T_{30}^{AVG}$  denotes a composite event of an average of the temperature during 30 minutes.  $(T_{room1} + T_{room7})_{30}^{AVG}$  denotes to take an average of room 1 and 7.

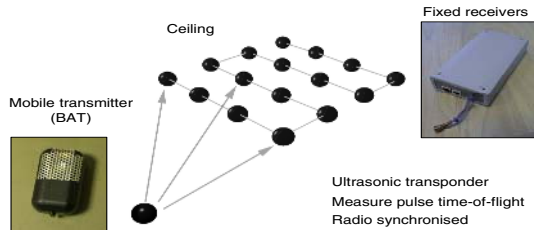
### 4 Prototype with the Active BAT System

Sentient computing is a type of ubiquitous computing which uses sensors to perceive its environment and react accordingly. A use of the sensors is to construct a world model, which allows location-aware or context-aware applications to be constructed. One research prototype of a sentient computing system was the work at AT&T Laboratories in the 1990s and the research continues at our



Computer Laboratory by means of the Active BAT system [2]. This is a low power, wireless, indoor location system accurate up to 3cm. It uses an ultrasound time-of-light trilateration technique to provide accurate physical positioning.

Users and objects carry Active BAT tags. In response to a request that the controller sends via short-range radio, a BAT emits an ultrasonic pulse to a grid of ceiling mounted receivers. At the same time that the controller sends the radio frequency request packet, it also sends a synchronized reset signal to the ceiling sensors using a wired serial network. Each ceiling sensor measures the time interval from reset to ultrasonic pulse arrival and computes its distance from the BAT. The local controller then forwards the distance measurements to a central controller, which performs the trilateration computation. Statistical pruning eliminates erroneous sensor measurements caused by a ceiling sensor hearing a reflected ultrasound pulse instead of one that travelled along the direct path from the BAT to the sensor. The SPIRIT (SPatially Indexed Resource Identification



**Fig. 3.** Active BAT

and Tracking) [2] provides a platform for maintaining spatial context based on raw location information derived from the Active BAT location system. It uses CORBA to access information and spatial indexing to deliver high-level events such as 'Alice has entered the kitchen' to listening context aware applications. SPIRIT models the physical world in a bottom up manner, translating absolute location events for objects into relative location events, associating a set of spaces with a given object and calculating containment and overlap relationships among such spaces, by means of a scalable spatial indexing algorithm. However, this bottom-up approach is not as powerful in expressing contextual situations.

#### 4.1 Distributed Gateways

The current Active BAT system employs a centralized architecture, and all the data are gathered in the database, where computational power is cheap. The Active BAT system, as described, is expensive to implement in that it requires large installations, has a centralized structure. The centralized structure allows for easy computation and implementation, since all distance estimates can be quickly shipped to a place where computational power is cheap. Moreover, the active mobile architecture facilitates the collection of multiple simultaneous distance samples at the fixed nodes, which can produce more accurate position estimates relative to a passive mobile architecture.

It is inherently scalable both in terms of sensor data acquisition and management as well as software components. However, when constructing real-time mobile ad hoc communications with resource-constrained devices, a distributed coordination must be supported, so that mobile device users can subscribe certain information promptly. We simulate each room and corridors hold gateway nodes (see the location map Fig.4), which is capable to participate in event broker grids. The software design follows the service-oriented architecture described in Section 2. Thus, each local gateway node performs event filtering and correlation. Each local node registers the service that associates states with abstractions such as 'Andy in the room SN04'. These states are decomposed to the units

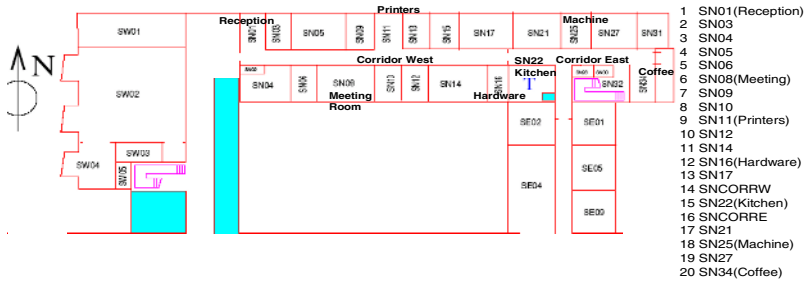


Fig. 4. Active BAT Location Map

operable by the event broker grid, where event correlation and aggregation and filtering are supported. The details of high-level language for service composition substantially event type definition is still under development, and out of scope of this paper. The used data is taken on May 20th in 2003 for 24 hours. The total number of original events received by the ceiling units is around 400,000, and a sample is shown in Fig.5. On average a sighting probably gets around 10 receptions, of which perhaps 2 will be 100% noise, 2 or so will be too noisy and will be rejected and the rest will be used for the final estimate. The format is:

```

----- Position Start
TIME: [02 0c 30 bb fa e5]
DARR: 2 1000.582092 1044.230957 2.320667 31052.382812 1.302316 1 -
DARR: 22 999.148926 1043.030518 2.319667 4677.762695 2.356863 1 -
DARR: 23 999.549744 1044.223877 2.319667 2388.645020 2.217386 1 -
DARR: 24 999.149475 1045.423706 2.323667 4777.290039 1.539556 1 -
DARR: 24 999.149475 1045.423706 2.323667 3383.913574 2.123533 2 -
Temperature: 27Curtailed: 0
RESULT: 0 1000.422546 1045.085571 1.182180 0.673943 1.631099 1.966668 0.511598 00 11
TIME: (UNIX TIME in hex)
DARR: (Receiver chain)(Rec x pos)(Rec y pos)(Rec z pos)(amplitude)(range)(set)(state)
RESULT: (error flag)(x)(y)(z)(error)....
    
```

Fig. 5. Active BAT Raw Events

The 'set' value can be 1 or 2 and represents whether the pulse was the first received or the second (so the pulses marked 2 are irrelevant to positioning, but there for other uses). A '1' pulse can be assigned a state A (accepted and used in the positioning calculation) or R (rejected and not used). After the position calculation, the total number of events around 200,000 are created (see Fig.6). This shows BAT data after the location of the user is calculated, which consists of timestamp, user, area, coordination (X, Y, Z) and orientation.

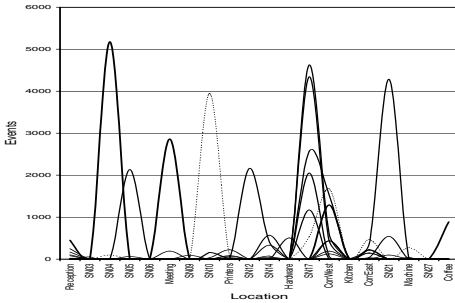
```

30408.802618,10,SN09,1002.335327,1033.320801,1.261441,-22.443605
30408.856115,10,SN09,1002.520386,1033.289429,1.251856,-20.335649
30409.063099,10,SN09,1002.533203,1033.279297,1.285185,-20.326197
30409.112594,10,SN09,1002.732910,1033.234863,1.270585,-22.712467
30409.315079,10,SN09,1002.921448,1033.175903,1.271525,-54.598316
30409.370575,10,SN09,1002.994690,1033.126587,1.283121,-56.499645
30409.564561,10,SN09,1003.170227,1033.044556,1.298443,-52.581676
    
```

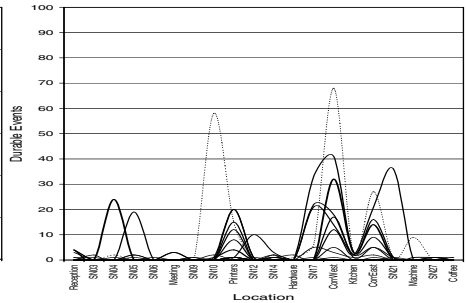
**Fig. 6.** Active BAT Location Events

### 4.2 Experiments

We performed several event correlations, and among those, we show the use of durable events below. Fig.7 depicts the number of events over the local gateway nodes without durable event and Fig.8 shows the same operation with durable event compositions. During this experiment, 21 BAT holders participated. The result shows dramatic reduction of event occurrences by the use of durable events, where the state of the target object is maintained.

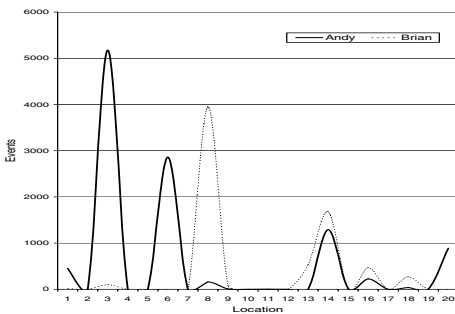


**Fig. 7.** Events over Locations

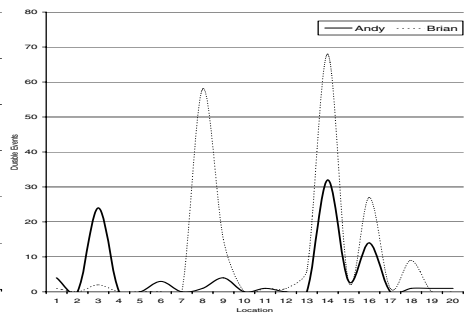


**Fig. 8.** Durable Events over Locations

Fig.9 and Fig.10 depict the events identified on the BAT holders Andy and Brian. Andy’s office is most likely the location 3 (room SN04), where the highest number of events is recorded. Brian’s office is the location 8 (room SN10), where



**Fig. 9.** Events over Locations



**Fig. 10.** Durable Events over Location

also the large number of events is produced. See the numbers corresponding to the location is described in Fig.4. Fig.9 and Fig.10 show the events over the location, however they do not indicate when they occurred.

Fig.11 and Fig.12 depict the events over the timeline (24 hours). Most activities are recorded during the day time. Durable events composition over the timeline (24 hours) shows significant reduction of the number of events.

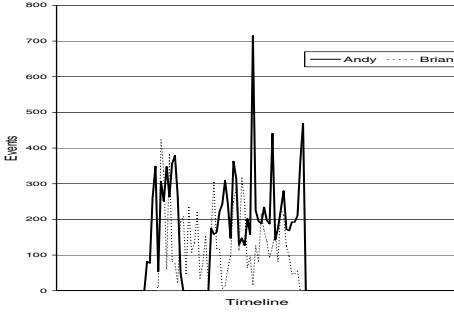


Fig. 11. Events over Time

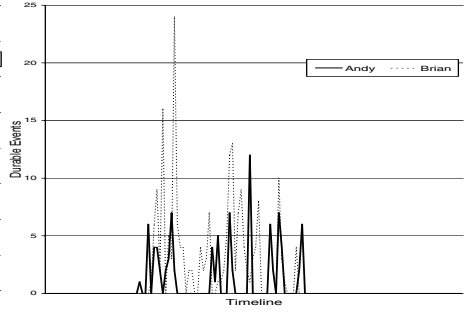


Fig. 12. Durable Events over Time

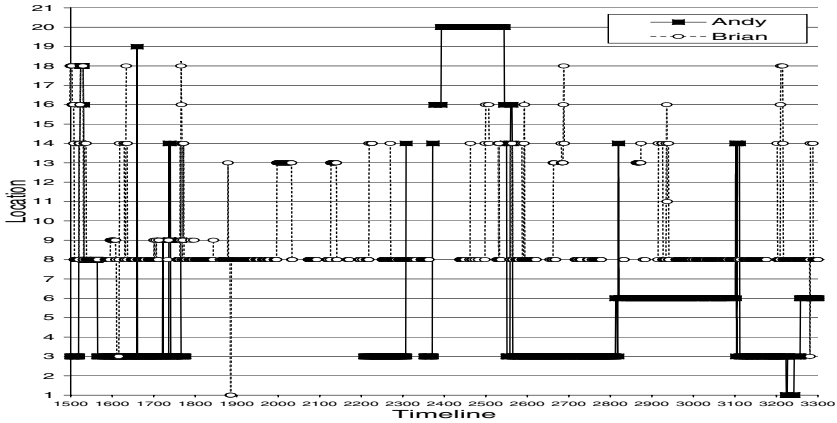


Fig. 13. Events over Location and Time

Fig.13 traces Andy and Brian over the time and location between time unit 1500 and 3300. One unit is 15 seconds, and 7 hours and half duration of activities is shown. It looks like Andy and Brian spent a lot of time in the location 8 (room SN10), where Brian's office as well at the both corridors.

Fig.14 shows the specific period, when they were positioned at the corridor west. The composite event  $(Brian;Andy)corrwest$  is detected at time unit 2564.

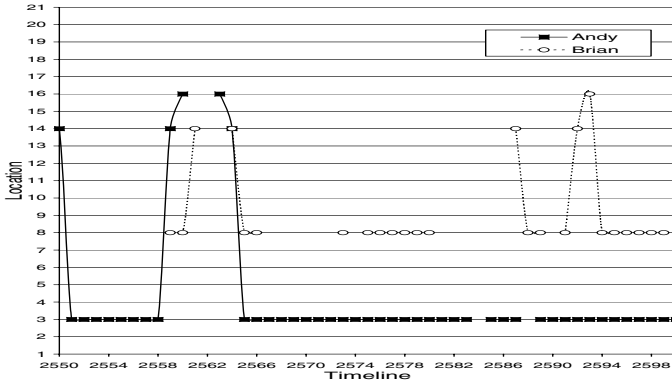


Fig. 14. Composite Event - (Brian;Andy)corrwest

In Fig.15, the detection of composite event  $(Andy+Brian)SN25(machine)$  is shown at the time unit between 1523 and 1529. A local gateway can detect this correlation, if the composite event is subscribed through the event broker. This composition could be a part of services provided by the service grid.

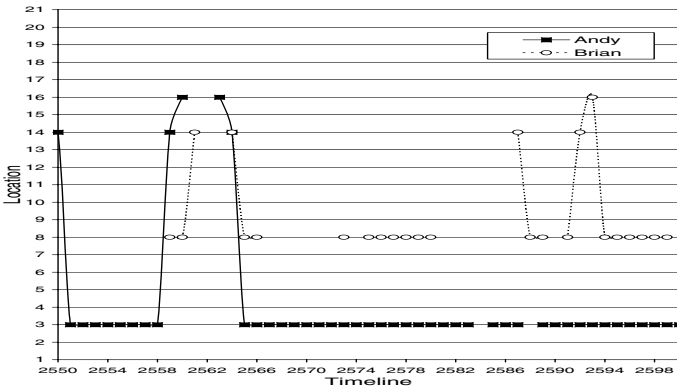


Fig. 15. Composite Event - (Andy+Brian)machine

### 4.3 Temporal Ordering in Active BAT System

The applications derived from Active BAT have high accuracy and real-time tracking requirements. Problems of time synchronization and coordination amongst beacons are easily resolved, because these systems are wired and have a centralized controller. The timestamp is derived from a Global Clock Generator (GCG), which is a hardware clock that sends ‘ticks’ to every component of the system over a serial link. When a location is computed, the location message is timestamped using the GCG. In general, GCG delay is in the order of microseconds, and the slowest part of the system is the bit that waits for ultrasound to propagate (speed of sound) after a position is requested but before a

position can be calculated. This delay is used to measure the distance between the BAT and the receiver at the ceiling. Once the location is calculated, the message then has to travel up to SPIRIT (order of milliseconds), and the event will be generated. However, no reliable information on that delay is considered. However, when gateways are distributed temporal ordering of events requires more complex time synchronization. The implementation of temporal ordering mechanism described in [7] is in progress. The current experiment assumes that all timestamps are properly synchronized.

## 5 Conclusions and Future Works

For WSN applications, distributed programming abstractions and middleware will be key technologies. In this paper, we introduce a middleware architecture and present an experimental prototype of object tracking with real world data produced by the Active BAT system. The tracking system uses the proposed middleware, which provides an event correlation service. Our integrated approach of event correlation, filtering and aggregation can express the complex composite event for tracking conditions. Event management will be a multi-step operation from event sources to final subscribers, combining information collected by wireless devices into higher-level information or knowledge in a global computing environment. Event broker grids should seamlessly disseminate relevant information generated by deeply embedded controllers to all interested entities in the global network, regardless of specific network characteristics, leading to a solution to construct large distributed systems over dynamic hybrid network environments. We are working on a complete implementation including various timestamping environments and different communication protocols.

**Acknowledgment.** This research is funded by EPSRC (Engineering and Physical Sciences Research Council) under grant GR/557303. We would like to thank Robert Harle (Digital Technology Group in University of Cambridge) for providing the Active BAT data.

## References

1. Ferscha, A. et al. A Light-Weight Component Model for Peer-to-Peer Applications. *Proc. MDC*, 2004.
2. Harter, A. et al. The Anatomy of a Context-Aware Application. *Proc. MobiCom*, 1999.
3. Madden, S. et al. TAG: A tiny aggregation service for ad-hoc sensor networks. *Proc. of Operating Systems Design and Implementation*, 2002.
4. OGSA Working Group, <http://www.ggf.org/ogsa-wg/>.
5. OSGi, <http://www.osgi.org>.
6. Reconfigurable Ubiquitous Networked Embedded Systems . <http://www.ist-runes.org>, 2005.
7. Yoneki, E. and Bacon, J. Unified Semantics for Event Correlation over Time and Space in Hybrid Network Environments. *Proc. CoopIS*, 2005.
8. Zhang, W. et al. Optimizing Tree Reconfiguration for Mobile Target Tracking in Sensor Networks. *Proc. Infocom*, 2004.

# Design of Integrated Routing System for Mobile Ad Hoc Networks Overlaying Peer-to-Peer Applications

Yan Annie Ding and David Everitt

School of Information Technologies,  
University of Sydney, NSW 2006, Australia  
{annie, deveritt}@it.usyd.edu.au

**Abstract.** This paper proposes a novel distributed routing system for integration between peer-to-peer (P2P) applications and mobile ad hoc networks (MANETs). This system takes advantage of the “zone” concept to reduce the multicast flooding. Significantly, the system investigates a mechanism to integrate key lookup in the application layer with routing in the link layer. The routing method provides a general-purpose technique that is not limited to any specific P2P applications. This paper presents the design of the routing system and sketches the layered architecture built according to the system functions.

## 1 Introduction

Peer-to-peer (P2P) overlay networks are self-organizing virtual networks over the IP-layer normally with TCP/IP connections. P2P services contribute to effective file sharing and resource locating in a computing environment without a central server. Mobile ad hoc networks (MANETs) are defined as wireless multi-hop networks formed dynamically and self-organized by mobile services for some purpose. Both of the networks have common characteristics in dynamicity, autonomy and “multi-hop” routing. These similarities motivate us to exploit the synergy [1] between P2P networks and MANETs.

P2P networks need high bandwidth network transmission and fast calculation speed, which can be fulfilled by the wired Internet. There are two main searching algorithms based on flooding [2] or Distributed Hash Table (DHT) [3], [4], [5], [6] in wired networks. These advanced P2P techniques are largely restricted in MANETs due to the autonomous distribution, scarce bandwidth and changing topology in physical links. The broadcast or flooding methods applied in the application layer are unable to supply smooth P2P applications under a MANET environment. The DHT based algorithm may induce excessive load. The severe situation is that the evoked broadcast storms [7] may break the whole system down when these wired methods are directly applied to MANETs.

With the growing demand in wireless Internet services and P2P applications such as instant messaging, ubiquitous computing and resource sharing, effective

P2P searching is very important, considering especially the vulnerable characteristics of MANETs. Although P2P systems and MANETs have many similar characteristics, they are completely different networks based on different layers and produce different broadcasts. When applying similar self-organizing P2P applications to MANETs, the major problem is how to efficiently query and find the matching documents in the MANET environment with limited power, scarce bandwidth and changing topology. We consider the characteristics of MANETs and propose a novel realistic system - Multicast Key-based Zone Routing (MKZR) based on practical multi-hop flooding. We aim to build P2P applications over MANETs smoothly and exploit the synergy between them.

In this paper, we sketch the system architecture and present the design model for the MKZR system. Section 2 analyses the current problems and the corresponding MKZR system idea. Section 3 describes our system architecture and algorithms in detail. Section 4 compares some related work. Section 5 gives some conclusions to our system designs and future work.

## 2 Current Problems and MKZR System

When applying P2P applications over “one-hop” MANETs with effective searching in one hop region, the method of flooding and caching contents [8] is a practical method. When overlaying P2P applications to larger MANETs, we can address some critical problems and the design notion for the MKZR system as follows:

### 2.1 Broadcast Storm

The broadcast storm problem in MANET was discussed in [7]. Here we refer to the broadcast storm problem from pure broadcast or flooding mechanism in P2P mobile ad hoc routing systems.

Pure P2P flooding is fragile due to the congestion after frequent query broadcasts. These query broadcasts are virtual broadcasts based on the application layer. Furthermore they can induce real broadcasts in the network layer. However the growing demands of P2P applications in MANETs aggregate a lot of text processing and querying, which unavoidably burden the network load in MANETs. The overhead in this network architecture is as high as  $O(N^2)$  [9]. This is disastrous in resource-scarce networks like MANETs. We reviewed current related works and found that systemic P2P routing algorithm design is still in its early stage in wireless MANETs. We argue that effective P2P routing mechanism is an important promising research area presenting the base for modern information sharing and ubiquitous computing under the P2P MANET environment.

In order to reduce the broadcasts, we use two novel and effective methods. Firstly, our MKZR routing system aims to integrate P2P searching and link layer routing. Secondly the MKZR introduces the “zone” concept to quickly locate the destination node.



## 2.2 Redundant Caching

Current caching of pairs <key, value> information in each node can reduce the querying time and hence result in effective use of network bandwidth. Caching systems in P2P MANETs obtain information mainly through dissemination and as a result each node caches the same contents of pairs. This method needs to be improved to save resources and reduce the administrative overhead.

The MKZR is motivated to be a routing system, which caches routing information instead of application layer keyword matching information <key, value>. Each node caches different routing information instead of the same content of keyword matches. In addition, our system will attempt to cache the routing information matches of <key, node address>. The node address belongs to the name space of link layer identification.

In the repository, an implementation of data index table presents the local data information indices.

## 3 Models and Algorithms

We propose the Multicast Key-based Zone Routing (MKZR) system based on the following main ideas:

**“Zone” Concept.** The network is divided into several zones where the hop distance between the central node and other nodes in the same zone is restricted to be within a specified radius, similar to Zone Routing Protocol (ZRP) [10]. The radius distance is calculated in hops and the peripheral nodes are situated around the radius distance. If a key search needs to find a new route, it will first multicast within the zone. If that fails, it will take advantage of the peripheral nodes to keep searching within their own zones. After the simulation, we will attempt to implement more functions such as storing more sharing information in peripheral nodes to reduce the searching overhead. We will compare the performance of the added functions.

**Routing Integration.** Our system aims to form a “key-based” whole system routing, which the key searching of the self-organized peers in the application layer can combine with the link layer routing in MANETs. The whole network is an integrated coordinating system to finish a key search or key lookup task. The integrating of P2P key searching and link layer routing of MANETs operates as a “resolver” in the network layer to transfer the IP address to the node ID in the link layer via a one-to-one mapping. The routing information stored in the application layer will be changed to <key, node ID route> instead of <key, IP address> at the last stage of the implementation. At the first stage the <key, IP address> is still used for the whole system simulation.

**Cache Content.** The cache system collects routing information as its content. There are two different tables in each mobile node: one is a routing table related with key <key, node ID> and the other is a neighbour table related with our zone division. The neighbour table is designed to reduce the broadcast overhead incurred from searching.

### 3.1 Layered Architecture

As P2P technologies mature in wired networks, P2P is considered as a decentralized symmetric computing model which various applications can run above. Our system aims to build a system model or protocol for a MANET running P2P applications. The mobile nodes in the model are randomly moving with P2P networks overlaid. The lower layer behaviour is “Make a move” or “Have a rest”, etc., as shown in Fig 3. The system architecture is layered as Fig. 1 according to our requirement analysis. The system is virtually divided as various subsystems or sub-models based on their different executing functions, e.g., the Issuer can send query or reply messages. The functions of each subsystem will be described in Section 3.3.

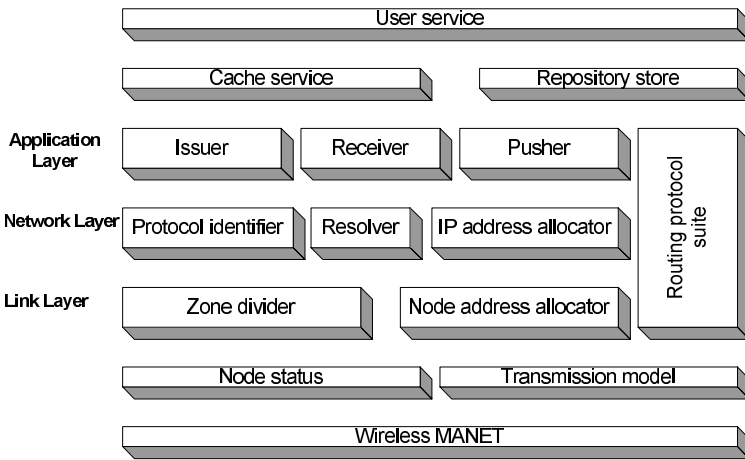


Fig. 1. Layered system architecture

### 3.2 Message Format

The system defines five types of message, *QUERY*, *REPLY*, *QUERY\_ZONE*, *REPLY\_NULL* and *PUSH* which are shown in Fig. 2.

### 3.3 System Function Overview

A key search starts from the application layer model of “Issuer”. The network layer model “Resolver” resolves the node ID in the link layer name space to an IP address. Protocol identifier can distinguish the different messages. Zone divider can divide zones according to the hop information in the protocol format. We list some main system functions in Fig. 3. We gradually describe our system model and functions from the application layer.

**QUERY:**

Type=0	Key	SRC	SEQ	RTT	TTL	R
--------	-----	-----	-----	-----	-----	---

**REPLY:**

Key	SRC	SEQ	RTT	TTL	DES=NULL	R
-----	-----	-----	-----	-----	----------	---

**QUERY\_ZONE**

Type=1	Key	SRC	SEQ	RTT	TTL	R
--------	-----	-----	-----	-----	-----	---

**REPLY\_NULL:**

Key	SRC	SEQ	RTT	TTL	DES=NULL	R
-----	-----	-----	-----	-----	----------	---

**PUSH:**

ID File	Index	DEX	R
---------	-------	-----	---

Fig. 2. Message format

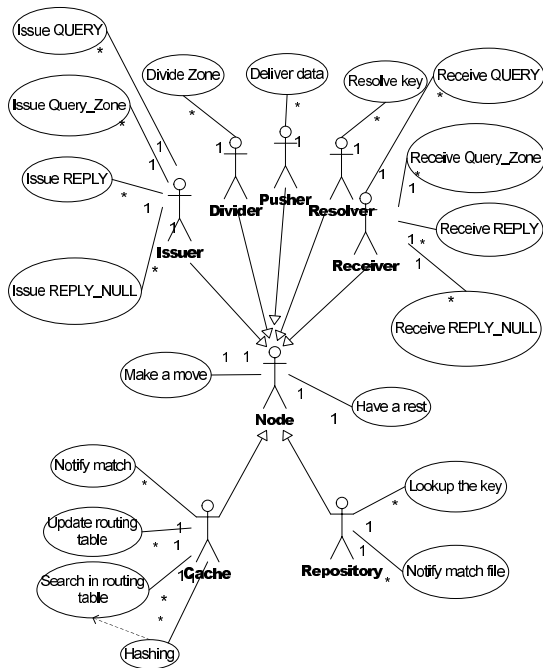


Fig. 3. System function overview

### 3.4 Application Layer

In our architecture model design, a routing task starts from a key search of a peer node in the application layer.

```

Peer (key, destination){
    key:=NULL;
    destination:=NULL;
    class Query();
    class DataReceive();
    Main( )
        key=k;
        destination=des;
        Query( );
        DataReceive( );
        do while (not stored)
        if (not cached) AND (not stored)
            destination:= Query(key).des;
            update cache;
            key:=NULL;
        else if cached AND destination < >NULL
            DataReceive(destination);
            update repository;
            destination:=NULL;
        endif;
    endif;
enddo;
}

```

**Query Algorithm.** MKZR provides an efficient key-searching algorithm by combining querying in the application layer with routing in the network layer. When a peer node issues a query for matching a key, it firstly checks its local index key table in Repository. If there is no match, it will search in the routing table. If it finds the key match in cache, the node directly sends a data request to the destination node and then is pushed back the data needed. If it does not find the route, the *QUERY* message piggybacks the key on the base of a link-layer broadcast message, which is distributed by link-layer flooding. The checking order is “local index key table - routing table - other nodes”. The *QUERY* message builds reverse paths to the issuer node in all the intermediate nodes it passed during the process to the destination node. In addition, when node multicasts a *QUERY* message there are two following situations need to be considered:

*Destination Node Within the Zone Area.* If there is a node that contains the content or owns the routing information of the key, this node will send a *REPLY* message containing the routing information such as the destination IP address (“DES” in message) of the key match directly to the issuer node. After the key-issuing node receives this *REPLY*, it will update its routing table and decide to stop the query. Those non-peripheral nodes without the key information are designed to simply discard the *QUERY* message. Those peripheral nodes without the key routing information will send a *REPLY* message without any

data included. We define this special *REPLY* message as *REPLY\_NULL* in Section 3.2. In our implementation, only peripheral nodes of a zone can send this empty data message. The issuer node discards this message after receiving *REPLY\_NULL* if it obtains the matching information from other nodes.

*Destination Node Outside the Zone Area.* The query task will hand over to peripheral nodes if there is no match within the zone. These peripheral nodes will keep multicasting within the scope of their zone after the issuer node sends a *QUERY\_ZONE* message. The notice must be sent with query key due to the last discarding operation.

**Data Delivery.** After the issuer node gets the routing information, it can build the direct connection to the destination node. The destination node can push the data to the issuer node after the issuer node request. The initial design ignores the information backup in P2P applications. The system can replicate some information requested frequently in some bone node such as peripheral nodes.

### 3.5 Network Layer and Link Layer

There is a “Resolver” in the network layer to complete the resolution between an IP address and a link layer node address. The implementation is important to the integrated routing between P2P networks and MANETs.

The protocol identifier model can extract the hop information. The system can then divide zones and implement neighbour node tables according to the hop information. The current P2P query algorithms or protocols are developed in wired networks which is not suitable to the MANETs as analyzed in Section 2. The MKZR link layer protocol is designed on the basis of multicast flooding but merged with the idea of Ad Hoc On-Demand Distance-Vector Protocol (AODV) [11] and our zone concept described above.

### 3.6 Integrated Routing

**Protocol Integration.** Firstly, the query algorithms in the application layer combine with link layer protocols. MKZR piggybacks the application layer key onto the message format and sends it through the link layer protocol to the next hop. This type of integrated routing like MKZR between P2P networks and MANETs operates a “key-based” routing style to drive the link layer protocol. Secondly, each message contains a field SRC (Source) for the unique identifier of the sender mobile node and a field SEQ for an increasing sequence number. The implementation of SRC and SEQ (Sequence) guarantees the forwarding flow and correct reverse path. The idea is borrowed from AODV.

**Name Space Integration.** In P2P routing, the general method to cope with the naming of key or related IP address is to hash them into one name space. The commonly used hash technique is consistent hashing Secure Hash Algorithm

(SHA-1) [12]. Our approach is to use pairs  $\langle \text{keyword}, \text{IP address} \rangle$  as the routing table entries and then hash them separately as  $\langle \text{key}, \text{peer ID} \rangle$ . We build a one-to-one mapping scheme between peer ID and link layer node ID. Furthermore, we will investigate the possibility of using the link layer node ID to hash as the peer ID in P2P routing tables.

As a result of the above integration, the virtual broadcast overlaid in the application layer can directly find the link layer route to reduce the algorithm complexity from  $O(N^2)$  to  $O(N)$ . In P2P MANETs, a key lookup can get to the destination node through the virtual application layer broadcasts to search the key and the network layer broadcasts to route the search information. The double broadcasts make  $O(N^2)$  routing steps in the pure flooding method. The MKZR protocol is a key-based application-layer driven network-layer routing protocol (We define this key-driven protocol). We propose to skip the virtual overlaying. This means that the virtual broadcasts induced by key searching are incorporated with the network layer routing broadcasts. If there is a pure key-based flooding method which is a network layer protocol like MKZR, then the performance of the pure key-based network flooding is  $O(N)$ .

## 4 Related Work

Wireless P2P research is often based on broadcast or flooding. For example, 7DS (Seven Degrees of Separation) [13] is implemented to exchange data for web-browsing among peers without connecting to the Internet. End users can advertise what they have, or query data objects related to the URL or host MAC addresses and get the information through broadcasts. 7DS is an approach not for discovery routing but for P2P application services.

Similar to 7DS, Passive Distributed Indexing (PDI) [8] and Optimized Routing Independent Overlay Network (ORION) [14] can provide file sharing locally by the implementation of a file routing table. Far more than that, PDI and ORION presents a P2P file sharing system based on MANET. Their general-purpose design may present the foundation for the later practical research in P2P MANETs. In particular, ORION introduces AODV techniques into the peer searching. ORION uses multicast flooding and stores file routing table and response routing table in the cache system for file sharing. However, data dissemination methods in both systems make the cache system in each node store the consistent content at the end.

Proem [15] is a general-purpose mobile computing platform for P2P applications and developments. It integrates P2P protocol, database and development tools. Messages are encoded using the XML language to communicate between peers. Proem does not consider how lower levels in MANETs are connected or self-organized. However, Dynamic P2P Source Routing (DPSR) [1] considers this point.

DPSR exploits the synergy between routing in P2P systems and that in MANETs, which designs a network layer routing protocol to integrate Pastry with DSR. However DPSR needs further investigation when considering the

seamless connection between application layer routing of Pastry and link layer routing.

The MIN architecture [16] proposes an evolving architecture overlaid on top of the application layer to integrate P2P applications with MANETs. The work is in its early stages.

## 5 Summary

Our proposal suggests a routing protocol especially for applying P2P applications over MANETs. We argue that the performance improvement for P2P MANETs is dependent on the breakthrough of MANET routing protocols. Our design attempts to target a general-purpose routing algorithm, which seamlessly integrates application layer key-lookup and network layer routing.

We have constructed a mobile model in ns-2 [17] and are implementing MKZR protocols and simulating the system in this environment. We will conduct the evaluation of the performance especially in packet delivery ratio (Data packets ratio) and routing overhead. In the future, our system attempts to supply an effective integrated routing model and protocols for MANETs running a variety of P2P applications

## References

1. Y.C. Hu, S. Das, and H. Pucha: Exploiting the Synergy between Peer-to-Peer and Mobile Ad Hoc Networks. Proc. HotOS-IX, 9th Workshop on Hot Topics in Operating Systems, Hawaii, May 2003, pp. 18-23.
2. The Gnutella Protocol Specification V0.4 (Document Revision1.2). [http://www9.limewire.com/developer/gnutella\\_protocol\\_0.4.pdf](http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf).
3. B. Zhao, J. Kubiawicz and A. Joseph: Tapestry: An Infrastructure for Fault-tolerant Wide-area Location and Routing. University of California Berkeley Technical Report, 2001.
4. A. Rowstron and P. Druschel. Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems. Proc. IFIP/ACM Int. Conf. on Distributed Systems Platforms (Middleware), Heideberg, 2001, pp. 329-350.
5. I. Stoica, R. Morris, D. Karger, F. Kaashoek and H. Balakrishann: Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications. Proc. ACM SIGCOMM 2001 - Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, San Diego, California, 2001, pp. 149-160.
6. S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker: A Scalable Content-addressable Network. Proc. IEEE/ACM SIGCOMM 2001 - Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, San Diego, California, 2001, pp. 161-172.
7. S. Ni, Y. Tseng, Y. Chen and J. Sheu: The Broadcast Storm Problem in a Mobile Ad Hoc Network. Proc. 5th Annual ACM/IEEE Int. Conference on Mobile Computing and Networking (MobiCom'99), Seattle, Washington, 1999, pp. 151-162.
8. C. Lindemann and O. Waldhorst: A Distributed Search Service for Peer-to-Peer File Sharing in Mobile Applications. Proc. IEEE Int. Conf. on Peer-to-Peer Computing (P2P), Linkping, Sweden, 2002, pp. 73-81.

9. G. Ding and B. Bhargava: Peer-to-Peer File-sharing over Mobile Ad Hoc Networks. Proc. Second IEEE Conf. on Pervasive Computing and Communications Workshops (PERCOMW), Orlando, 2004, pp. 104-108.
10. Z. Haas, M. Pearlman and P. Samar: Zone Routing Protocol - IETF Internet Draft. draft-ietf-manet-zrp-04.txt, Jan. 2001.
11. C.E. Perkins, E.M. Belding-Royer, and S. Das: Ad Hoc On-Demand Distance Vector (AODV) Routing. <http://moment.cs.ucsb.edu/pub/rfc3561.txt>, IETF RFC 3561, 2003
12. FIPS PUB 180-1, Secure Hash Standard. Federal Information Processing Standard Publication 180-1, National Institute of Standards and Technology, US Department of Commerce, Apr. 1995.
13. M. Papadopouli and H. Schulzrinne: Effects of power conservation, wireless coverage and cooperation on data dissemination among mobile devices. Proc. IEEE/ACM Int. Symp. on Mobile Ad Hoc Networking and Computing (MobiHoc 2001), Long Beach, California, 2001, pp. 117-127.
14. A. Klemm, C. Lindemann and O. Waldhorst: A Special-purpose Peer-to-Peer File Sharing System for Mobile Ad Hoc Networks. Proc. Workshop on Mobile Ad Hoc Networking and Computing, Sophia-Antipolis, France, Mar. 2003.
15. G. Kortuem, J. Schneider, D. Preuitt, T. Thompson, S. Fickas, Z. Segall. When Peer-to-Peer Comes Face-to-Face: Collaborative Peer-to-Peer Computing in Mobile Ad Hoc Networks. Proc. P2P2001, Linkping, Sweden, Aug. 2001.
16. L. Yan, K. Sere and X. Zhou. Towards an Integrated Architecture for Peer-to-Peer and Ad Hoc Overlay Network Applications. Proc. 10th IEEE Int. Workshop on Future Trends of Distributed Computing Systems (FTDCS'04), Suzhou, China, May 2004.
17. The VINT Project, UC Berkeley, LBL and Xerox PARC. The ns-2 manual - <http://www.isi.edu/nsnam/ns/ns-documentation.html>



# A Design of Privacy Conscious RFID System Using Customizing Privacy Policy Based Access Control

Byungil Lee and Howon Kim

Electronics and Telecommunications Research Institute, Korea,  
161 Gajeong-dong, Yuseong-gu, Daejeon, Korea  
{bglee, khw}@etri.re.kr  
<http://www.etri.re.kr>

**Abstract.** Recently, research related to RFID technology has progressed vigorously for the ubiquitous world. However, the privacy violation on tags embedded in products has become a major stumbling block. This paper suggests a new architecture for a user-friendly and secure RFID service, which uses customizing privacy policy based access control for collection or gathering of tagged information in an RFID information system. Proposals on feasible security mechanisms for low cost RFID systems and analyses of these mechanisms from both security and privacy points of view are also presented. The proposed architecture of an RFID privacy enhancing middleware system can expand the multi-domain level for various user services. Consequently, the proposed scheme effectively protects against abuse of tag related personal information using a consumer-configured privacy policy.

## 1 Introduction

Recently, low-cost radio frequency identification (RFID) has attracted growing interest from both industry and academic institutes. The RFID industry is very active, with numerous companies developing RFID tags of varying capabilities. With the growing importance of RFID for the ubiquitous infrastructure, numerous works on RFID technology including development of related applications have been conducted in a variety of fields. RFID can be potentially utilized in a broad range of everyday applications because of its capacity to readily relay information. Using RFID, the information of tagged products is given from an information server that utilizes data identified by a RF reader via the emission of an RF signal. While people often think of RFID tags as simply an updated replacement of the familiar bar code, they differ in several important ways, including usage of memory, identity individual items, writing of new data, interfacing with sensors, and digital data source. In order to adapt limited computing resource, such as RFID to secure sensor networks, new security approaches to cope with new threats have also been studied[1].

The areas where RFID technology can be applied are diverse, from inventory management and supply chain management to environmental surveillance and detection of intrusion in security systems. Using these kinds of management systems, an innovative business process can be constructed and various convenient services can be realized[2]. However, unless these systems are properly designed and constructed,

they can cause massive collateral damage to consumer privacy and company security. A store’s inventory labeled with unprotected tags may be monitored by unauthorized readers on behalf of competitors. In addition, personalized tags would radiate identifying personal information to any tag reader anywhere, thus making their every movement traceable. Despite RFID’s many advantages, there are many problems in deploying RFID technology extensively. The privacy violation of people carrying tag embedded items is one of the most serious problems in terms of expanding services utilizing this technology[3]. Consequently, in the present work, we address the privacy violation in the RFID system and investigate previously proposed algorithms for privacy issues.

With regard to the previously mentioned security threat to a store’s inventory several approaches to RFID security have been reported, including killing tags at the checkout, applying a rewritable memory, physical tag memory separation, hash encryption, random access hash, and hash chains[3-8]. These approaches are introduced in detail in the next section.

Based on the pervasive deployment of RFID tags, we propose a novel privacy-aware RFID architecture followed by a hierarchical privacy policy for customizing privacy preference in real-time.

The remainder of this paper is organized as follows. Section 2 gives a brief description of privacy problems and previously proposed algorithms for privacy protection. Section 3 delineates a model of privacy protection and a technical working mechanism using a hierarchical and customizing privacy policy. Finally, conclusions and a discussion of potential areas for future research are presented in Section 4.

## 2 Privacy on RFID and Analysis of RFID Security

### 2.1 Security and Privacy for RFID Service

Many companies and organizations, including EPC Global Network, have developed RFID systems and international standardization is ongoing. The architecture of an RFID system is shown in figure 1.

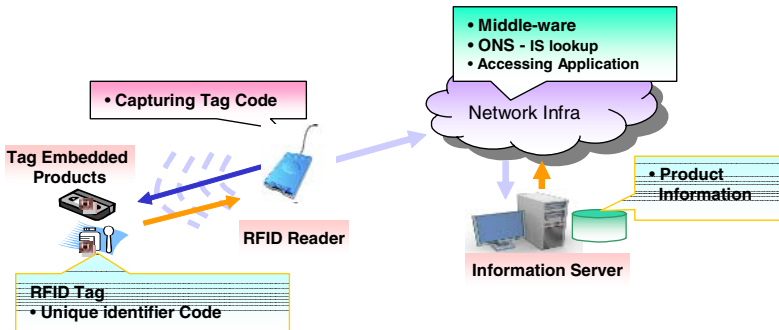


Fig. 1. RFID System Architecture

In this model, the user holding the RFID reader requests information from the tag attached to the product and the reader transmits this information to the back-end information server via middle-ware. The information server subsequently returns the product information to the reader and displays it to user. The service model, as shown in Figure 1, is comprised of a RFID Tag, a reader, middleware, and an information server.

In view of information protection, a serious problem for the RFID service is a threat to privacy. Here, the privacy issue involves the risk of exposing information stored in a tag and the leakage of information including personal data related to the carrier of the tagged products. Privacy protection in a RFID system can be considered from two points of view. One is privacy protection between the tag and the reader in this regard, ID encryption, prevention of location tracking, and countermeasures against tag forgeries are utilized. The other is the exposure of information stored in the information server along with tagged items. First, we consider the exposure of information between the tag and reader, and then discuss the proposed privacy mechanism.

## 2.2 Previous Works on Smart Tag Based Security of RFID for Secure Communication

The previously proposed algorithms for the protection of traceability are Hash-Lock, Randomized Hash Lock, Re-encryption and Hash-Chain[3-9].

### 2.2.1 Randomized Hash Lock

Randomized Hash Lock algorithm improved the problem of traceability that the Hash-Lock algorithm has because of the unchanged meta-ID. Tag has two different hash functions and RNG (acronym for random number generator). The algorithm is following.

- 1) Tag generates a random number R using RNG.
- 2) Tag computes the Hash function  $H(x)$  using the result of concatenation( $\parallel$ ) of tag ID and R as input and gets hash value V,  $V = H(\text{ID} \parallel R)$
- 3) Tag sends V and R to Reader and reader sends them to the database server storing tag ID. The server calculate the hash function using the result from concatenating the received R and tag ID stored in database and compares the calculated result with the received V from reader. If the server finds the same value with V, server returns tag ID back to reader.

This algorithm can solve the privacy violation by generating a different random number every challenge. However, there is a problem that the database server has to traverse sequentially all records to find a matched tag ID.

### 2.2.2 Re-encryption

A re-encryption algorithm can avoid traceability since it periodically updates and encrypts the stored data on a tag's memory space. However, due to the restricted resources of a passive tag, the data on the tag cannot be updated by the tag itself. Therefore, the data is encrypted by another party outside of the tag. Even though traceabil-

ity can be prevented by changing the data on the tag memory, additional equipment should be made available.

### 2.2.3 Hash-Chain

In this algorithm, tag embeds the two different hash functions  $G(x)$  and  $H(x)$  and stores the initial state information  $S1$ . As the response to the reader, tag sends  $a1$  which is the result of calculating the hash function  $G(S1)$ ,  $a1 = G(S1)$

The tag's secret information  $S_i$  is altered into the value acquired as the result of the computing  $H(S_{i-1})$ ,  $S1 = H(S_{i-1})$

The Back-End database server stores the pair of tag ID and  $S1$ . The reader sends  $a_i$  received from the tag to the database server. The server finds the tag ID which results the identical value with  $a_i$  by computing hash function using  $S$  as input.

$$S1 = G(H_i(S1))$$

By these two different hash functions, there are benefits that the privacy of the owner can be protected and the tag can work with low power. However if the tag information is exposed, there are a few problems such as traceability and the forgery of the tag.

## 2.3 Privacy Protection for Personalized RFID Tag

Passive RFID systems, as opposed to smart active tags, are the most promising approach to provide low-cost ubiquitous tagging capability with adequate performance for most supply chain management applications. However, due to limitations in resources and computing capability in a low-cost RFID tag, a smart security mechanism is not applicable[10]. As highly coupled limited resource problem of a low-cost passive RFID tag, security and privacy technologies (reader, middleware and server side) can be added or enhanced to overcome the security weaknesses in an RFID system.

As tags are more widely utilized, the amount of information around us will increase dramatically, thus posing a critical privacy problem[10]. Privacy can be defined as the right that personal information be protected from exposure to others. From this point of view, we define requirements of privacy protection in an RFID system.

- Avoid collecting unnecessary private information in the RFID system
- Employ a controllable access control mechanism to the collected data in the RFID system
- Avoid collecting unnecessary private tracing in the RFID system
- Secure communication should be established for all communication links with the RFID system

Using the code acquired from the tag, the accessing application can access information about the tagged product by querying the server. In order to ensure privacy in this process, a mechanism that encrypts and decrypts the outgoing data from tag and server has been proposed. However, this creates limitations in the applicability of the RFID service. In addition, if decryption of the tag data is successful, all information can be exposed. For this reason, we propose a method that protects privacy in the RFID system using a personal privacy policy in order to administer information more flexibly and securely as well as mitigate the aforementioned problems.

### 3 Proposed Privacy Conscious RFID System

The way which protects privacy through implementing Privacy Preference is described like the following: the technology that provides pre-defined information in the level tag owner has selected and set according to his (or her) preference.

#### 3.1 The Architecture for Personal Privacy Policy

Our proposed architecture is uniquely identified by the attachment of an RFID tag. Figure 2 shows the fundamental architecture for the privacy conscious RFID system proposed in this paper. The components of this architecture, as depicted in Figure 2, are (i) Tag and RFID Readers, (ii) Middleware, (iii) Information Servers, (iv) a Privacy Manager, and (v) a Privacy Policy Manager.

The tag owner set up his (or her) privacy policy for the tag and configures the authority of access and the related privacy level for sharing of information through a terminal in the Privacy Manager. The tag sends the pre-established privacy value when the reader requests it and the reader sends it back to the back-end information server. The server analyzes the data received from the reader and provides information in accordance with the privacy level set previously.

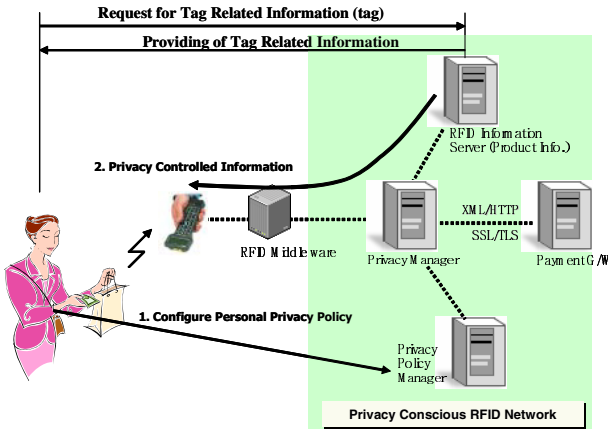


Fig. 2. Architecture of Privacy Protection with a RFID system

In this paper, we discuss an information system that allows access of information at the right place and time with low communication cost through a user privacy profile. The proposed system employs the following procedures:

- 1) Introduction of an information management table based on the user configured privacy and access policy
- 2) Provision of privacy controlled tag related information if a user requests information inform a personalized tag.



level 3 for an access group third party(others), the information of product class(such as health, food, cloth etc) can only provide to others. The associated table access policy with privacy level is needed for the following purpose in the policy database of the privacy manager. When the third party queries the information of tag, the owner wants to know who contact the database to obtain the tag related information. And owner wants to make the decision whether provide or not the information of the tagged product. For example in table 2, access group 3 mapped to the privacy level 3 and access group 4 mapped to the privacy level 2. Owner of product can select privacy level for access group.

### 3.3 Privacy and Access Policy of RFID by Product

The user memory bank of RFID tag can store the encrypted data of privacy policy owner configured, as shown in Figure 3. For example in Table 2, access control policy is 4,4,3,2,1 and the data is stored in tag. When anyone read the tag, encrypted the policy data must be transport to server securely and the information of tag can be provided by the access policy.

The tag memory is insecure and susceptible to physical attacks. This includes a myriad of attacks such as shaped charges, laser etching, ion-probes, tempest attacks, clock glitching, and many others[10]. Fortunately, these attacks require physical tag access and are not easily carried out in public or on a wide scale without detection[10-13].

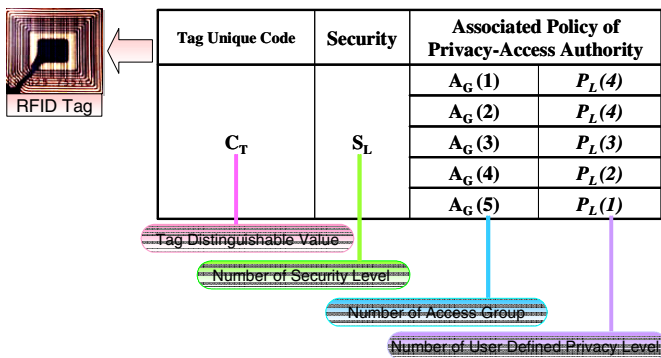


Fig. 3. Data Structure of Tag for Privacy and Security

The security level is a key factor to ensure the privacy policy and related data along with the tag, reader, middleware, and server. Furthermore, it can be an alternative to solve a request for information by maliciously changing the privacy level. Table 3 represents the additionally required phases according to the request. A variety of access control methods can be added as the importance of the information increases. The enforcement of the access control should be proportionally complicated and stronger according to the security level.

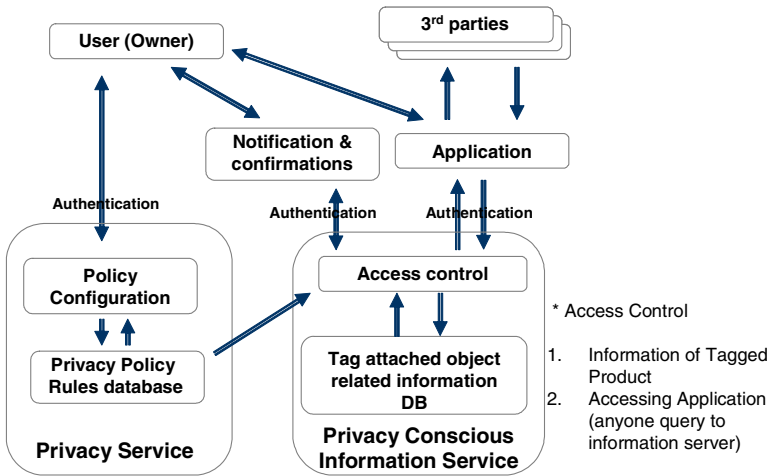
**Table 3.** Security Level for Secure Communication in RFID system

Security Level	1	2	3	4
No Security	o	x	x	x
Symmetric key based Security(password based lock/unlock)	x	o	x	x
Symmetric based Security(encryption of Tag code), Protection of Location based Traceability	x	x	o	x
Public Key based Encryption, Protection of Location based Traceability	x	x	x	o

**3.4 Privacy Protection Mechanism for Consumer Privacy**

The information about a product can transfer as a XML format with security(WS Security) and also can be subject to access controlled by a standard technology such as XACML.

The procedures to set the policy for the disclosure of the product information and processing by the information server are shown in Figure 4.



**Fig. 4.** RFID Privacy Protection Mechanism in Information Server

- 1) The company’s privacy manager receives the privacy rule and policy from owner. They are the information defined in the tag and server with the owner’s privacy, security and access policy.
- 2) The accessing application requests tag attached product information with the tag code and the security and privacy policy data to the information server.
- 3) The information server parses the user request.
- 4) The privacy manager make decision the final privacy level using the privacy and security level in the tag’s user memory, the user’s previous policy stored in the



server, and the information(who, what propose etc) of accessing user and company’s privacy rule.

$$\begin{array}{l}
 P_{\text{server}} : \text{Policy defined in Policy Server} \\
 P_{\text{tag}} : \text{Customized Policy in Tag} \\
 [P_{\text{tag}} \oplus P_{\text{server}} \oplus P_{\text{company}}] \text{ with access policy} \rightarrow P_{\text{out}} \rightarrow \text{Server Decision}
 \end{array}$$

- 5) If the server needs to notify to the owner by notify policy to make decision access policy, the server notifies and polls via a mobile SMS message.
- 6) If the user replies with a positive response and the response is authenticated, the server provides the information according to the authority previously configured by the tag owner.

The Information server stores the event logs and notifies them to the owner. Through monitoring the notified logs, the owner can re-configure the privacy preference and the access control for optimized privacy protection.

### 3.5 Hierarchical Structure of Privacy Domain for RFID Privacy

In this section we discuss the structure for the RFID privacy domain, which provides a personal policy based global combination of the RFID network.

First, our privacy domain architecture is hierarchical in nature, organized as a balanced tree topology, as shown in Figure 5.

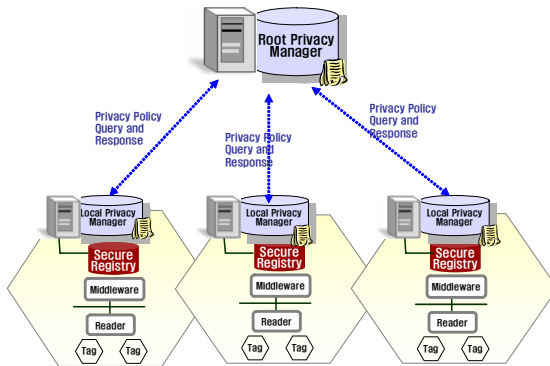


Fig. 5. Hierarchical Structure of RFID Privacy Domain for User’s Privacy Protection

The root server of the privacy policy sets up the personal preference policy according to service category. The low positioned local privacy manager inherits the root’s policy and adapts the policy to its own products. The local manager is audited by the root manager in terms of whether the inherited policy is properly adapted and whether the user’s policy as been maintained properly. In hierarchal structure, PPP (Privacy Policy Protocol) is used between the root privacy manager and the local privacy manager for exchange of policy. The root privacy manager establishes a permitted limit, which is also used for application to the local domain policy.

The messages of PPP consist of Query Message, Reply Message, Privacy Policy Message, Ack Message, Transfer Message, and Keep Alive Message.

In the hierarchal structure, the user can configure a customizing policy to root the privacy manager. Then, the policy is applied to the local privacy manager for local domain RFID privacy using PPP.

## 4 Conclusion

In this paper, we considered privacy protection for the owner of a tagged product in terms of two aspects: privacy between the tag and reader and exposure of the product information. The proposed algorithms were presented in terms of privacy policy whereas we introduced a privacy protection scheme using owner configured privacy preference policy. In the proposed mechanism, the owner of the tagged product sets the privacy level and security level on tag data fields and configures the level of the information that will be accessible to the public and the access control. By doing so, customized privacy protection can be realized. In this regard, the suggested mechanism is an effective solution for privacy protection in an RFID system.

## References

1. Seunghun Jin, et al.: Cluster-based Trust Evaluation Scheme in Ad Hoc Network. *ETRI Journal*, Vol.27, No.4 (2005)
2. S. E. Sarma, S. A. Weis, and D.W. Engels. RFID systems, security and privacy implications Technical Report MIT-AUTOID-WH-014, AutoID Center, MIT (2002)
3. Weis, S. et al. "Security and Privacy Aspects of Low-Cost Radio Frequency identification Systems", First Intern. Conference on Security in Pervasive Computing (SPC) (2003)
4. P. Golle, M. Jakobsson, A. Juels, and P. Syverson. Universal re-encryption for mixnets (2002)
5. M. Ohkubo, K. Suzuki and S. Kinoshita, Cryptographic Approach to "Privacy-Friendly Tags, RFID Privacy Workshop2003 (2003)
6. Jan E. Hennig, Peter B. Ladkin, Bern sieker, Privacy Enhancing Technology Concepts for RFID Technology Scrutinised, RVS-RR-04-02, 28 October (2004)
7. Ari Juels, Ronald L Rivest, Michael Szydlo "The Blocker Tag: Selective Blocking of RFID Tags for Consumer Privacy" 10th ACM Conference on Computer and Communications Security (2003)
8. Ari Juels, Ravikanth Pappu, Squealing RFID-Enabled Banknotes In R. Wright, ed., *Financial Cryptography* (2003)
9. Simson Garfinkel, Beth Rosenberg, *RFID Applications, Security, and Privacy*, Addison-Wesly (2005)
10. Stephen A. et al., Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems, *Security in Pervasive Computing 2003*, LNCS 2802, (2004) pp. 201–212
11. Sanjay E. Sarma, et al., *RFID Systems and Security and Privacy Implications*, CHES 2002, LNCS 2523, pp. 454–469, (2003).
12. Dirk Henrici and Paul Müller, Tackling Security and Privacy Issues in Radio Frequency Identification Devices, *Pervasive 2004*, LNCS 3001 (2004) pp. 219-224

# Efficient Resource Management Scheme of TCP Buffer Tuned Parallel Stream to Optimize System Performance\*

Kun Myon Choi<sup>1</sup>, Eui-Nam Huh<sup>2</sup>, and Hyunseung Choo<sup>1,\*\*</sup>

<sup>1</sup> School of Information and Communication Engineering,  
Sungkyunkwan University, Korea  
{dilly97, choo}@ece.skku.ac.kr

<sup>2</sup> College of Electronics and Information, Kyung Hee University, Korea  
johnhuh@khu.ac.kr

**Abstract.** GridFTP is a high-performance, secure and reliable parallel data transfer protocol, used for transferring widely distributed data. Currently it allows users to configure the number of parallel streams and socket buffer size. However, the tuning procedure for its optimal combinations is a time consuming task. The socket handlers and buffers are important system resources and must therefore be carefully managed. In this paper, an efficient resource management scheme which predicts optimal combinations based on a simple regression equation is proposed. In addition, the equation is verified by comparing measured and predicted values and we apply the equation to an actual experiment on the KOREN. The result demonstrates that the equation predicts excellently with only 8% error boundary. This approach eliminates the time wasted tuning procedure. These results can be utilized directly and widely for the fast decision in typical applications such as GridFTP.

## 1 Introduction

According to the recent research and development on global climates, high energy physics, and computational genomics, Grid technology has advanced as a promising method for widely distributed high speed data transfer. The Grid Alliance Team developed GridFTP especially to support safe and efficient transfer of distributed data. In addition, this tool uses a parallel data transfer mechanism to enable application level high-speed data transfer. Currently, GridFTP allows users to configure the number of parallel sockets and socket buffer size. In other words, the decision, for selecting the number of parallel sockets and buffer size for maximizing the network availability, should be made by a user [1, 2].

So far, studies regarding high performance data transfer. TCP buffer tuning [3, 4, 5] and parallel data transferring [6, 7] have been executed separately. In a Gigabit scale network, the separate use of these transferring mechanisms can cause each

---

\* This work was supported in parts by Brain Korea 21 and the Ministry of Information and Communication in Republic of Korea.

\*\* Corresponding author.

end system overhead, due to its large buffer size or overwhelming parallel streams. To overcome this problem, using parallel streams simultaneously with TCP buffer tuning, can be a good solution. This combination balances the number of parallel streams and buffer size, reducing these to a reasonably low level.

The socket buffer size and number of socket handlers are directly related to the memory limits and maximum number of concurrent users, respectively. Therefore, such system resources need to be carefully and efficiently managed. However, balancing procedures require background knowledge in networks, and is obviously a time-consuming task. Actually, it often takes more than 20 minutes. In addition, only a few studies have delved into the characteristics of parallel data transfer. In particular, the subject of buffer tuned parallel sockets has never been examined in detail.

Therefore, in this paper, the characteristics of buffer tuning and parallel data transfer are discussed by analyzing the results of various experiments. After, a simple relational equation, representing the relationship between the optimal number of parallel streams and buffer size, is developed. The system validity is also verified on the KOREA advanced REsearch Network (KOREN) testbed that supports Giga-bit bandwidth, by applying the equation to actual data transfer.

The remaining sections of this paper are organized as follows. Section 2 explains the concept of manual tuning, automatic tuning, parallel data transfer mechanism, and multiple regression. Section 3 develops a simple relational equation based on the analysis of various experiments. Next we verify the accuracy by applying the equation to an actual data transfer demonstration on the KOREN. In Section 5, this paper is concluded, and future work is discussed.

## 2 Related Works

Current main streams high performance data transfer mechanisms are divided into two categories; the first is a parallel data transfer mechanism [6, 7] to avoid TCP buffer limit by using multiple sockets, and the second is TCP buffer tuning, which adjusts the socket buffer to an appropriate size for each connection. These TCP buffer tuning can also be categorized into manual turning [3, 4] and automatic tuning [5]. The former configures the optimized buffer size referred to

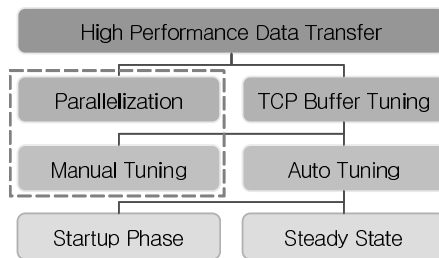


Fig. 1. Hierarchy of high performance transfer mechanisms

the network estimation tool by a network administrator, and the latter tunes up without the network administrator, through tuning daemon tool or the tuning algorithm in TCP stack. Automatic tuning techniques are also classified into startup phase tuning [8] which configures the buffer size at connection setup time, and the steady state tuning [5, 9, 10] which continuously adjusts buffer size during the life time. This paper focuses on both the parallel transfer mechanism and TCP buffer tuning represented by the dotted line in Figure 1.

**Manual Socket Buffer Tuning:** The throughput of FTP on links having both High Bandwidth and High Delay (HBHD) can be improved by “tuning” operating system parameters, using the TCP networking stack on the file transfer endpoints. This tuning usually involves increasing memory buffer sizes to fulfill the large data window’s request in high bandwidth. TCP uses the concept of “windows” to throttle the amount of data that the sender injects into network, depending on the receiver’s available capacity and the other link traffic. A window is the amount of new unacknowledged data that the TCP allows to be in flight between sender and receiver. In order to continuously full the link capacity, the TCP must maintain the window size into the Round Trip Latency (RTT) multiplied by the bandwidth, therefore, called the Bandwidth-Delay Product (BDP). Generally, in order to achieve maximum throughput, TCP requires a receive buffer equal to or greater than the BDP, in addition, TCP often requires a sender side socket buffer of  $2 \times \text{BDP}$  to recover from errors [3, 4].

**Automatic Socket Buffer Tuning:** Linux autotuning refers to a memory management technique used in the Linux kernel, (stable version 2.4). The central difference between Linux autotuning and other automatic tuning techniques is that autotuning does not measure the BDP, therefore it does not consider the condition of the network. Linux autotuning increases window size for TCP connections when the socket buffer is full. Therefore the performance enhancements represent a side effect of increased socket buffer size. The range of memory, which can be allocated to the socket buffer, is limited by the proc files `tcp_wmem`, `tcp_rmem`.

**Parallel Data Transmission:** Parallel data transmission establishes multiple connections; data is split and transferred simultaneously through each connection. This technique operates at application level. GridFTP uses this technique, because it can complement the shortcomings of TCP buffer tuning, and simultaneously can achieve maximum throughput on high performance-long distance network environments [2, 6, 7]. Parallel data transferring, unlike buffer tuning, can get maximum achievable throughput even if RTT is short by using a sufficient number of streams. For this reason, it is a good complementary scheme for the buffer tuning. To identify the strength and weakness of parallel data transferring, well-known facts regarding parallel streams are summarized below.

- Maximum throughput can be achieved without the administrator’s help.
- Parallel streams are stronger than the buffer tuned stream against packet loss.
- Parallel data transferring is effective regardless of RTT.
- Increasing the number of parallel streams reduces traffic interference.

**Multiple Regression:** The purpose of multiple regression is to identify the relationship between several independent variables and a dependent variable, and to predict dependent variable by using two or more independent variables. The multiple regression equation takes the form  $y = b_1x_1 + b_2x_2 + \dots + b_nx_n + c$ . The  $b_i$ 's are the regression coefficients, representing the amount the dependent variable  $y$  changes when the independent changes 1 unit. The  $c$  is the constant, where the regression line intercepts the  $y$  axis, representing the amount the dependent  $y$  will be when all the independent variables are 0. Power terms can be added as independent variables to explore curvilinear effects. Cross-product terms also can be added as independent variables to explore interaction effects. Interaction effects are sometimes called moderator effects because the interacting third variable which changes the relation between two original variables is a moderator variable which moderates the original relationship.

### 3 The Proposed Scheme

In this section, we propose an efficient resource management scheme that predicts optimal combinations of the number of parallel streams and buffer size with considering delay variation. The main idea of this scheme is that the relationship of the number of parallel streams, buffer size, and network delay can be represented by a single regression equation. In order to derive this equation, multiple regression approach is used. The important contribution on using this equation is that it can eliminate the time wasted for whole tuning procedure. In addition, this equation gives important information which is used for balancing the number of sockets and buffer size to tolerable level.

#### 3.1 Modeling of Regression Equation

In this subsection we first set up some notations which will be used throughout this paper. A certain buffer size exists, at which, increasing the buffer size has no effect on throughput. In addition, a certain number of parallel streams exist, at which, increasing the number of sockets does not increase throughput. In addition, there is tradeoff between the optimal buffer size and optimal number of parallel sockets. Therefore, It can be said that an optimal relationship exists for achieving maximum throughput without waste of memory or socket handler. For convenience, the buffer size and the number of parallel streams in its optimal relationship are denoted as  $B_o$  (Optimal Buffer size) and  $P_o$  (Optimal number of Parallel sockets), respectively.

In the case of buffer tuning, throughput increases proportional to the buffer size. However, the throughput is may not be exactly proportional to the buffer size. Therefore, a power term  $C_2$  is added as independent variable to Equation 1 below for representing curvature. In the case of parallel transfer, the  $i^{th}$  individual socket throughput is denoted by  $T_i$  and the aggregate throughput is denoted by  $T$ . The  $T$  is exactly proportional to the  $T_i$  when  $T$  is less than the maximum available throughput, and there are  $P_o$  number of buffer tuned

sockets. Therefore, the aggregate throughput is represented as a cross product of  $B_o$  and  $P_o$  terms. In addition, throughput increases when BDP increase. This proportional relationship can be represented by introducing a constant  $C_1$ . The equation for the collective characteristics of  $P_o$  based on  $B_o$  and RTT can be derived as follows:

$$BW \times RTT = C_1 \times P_o \times B_o^{C_2} \quad (1)$$

$$P_o = \frac{RTT \times BW}{C_1 \times B_o^{C_2}} = C'_1 \times \frac{RTT}{B_o^{C_2}} \quad (2)$$

Here, the bandwidth  $BW$  is fixed to  $1Gbps$ , because we assume that the network environment is Giga-bit Ethernet. For this reason,  $C'_1$  is introduced which is representing  $\frac{BW}{C_1}$ . The decision of correlation coefficients differs between experiments. Therefore, average coefficients of each experiments are used, and these can be determined using heuristic multiple regression analysis of many experiments results.

### 3.2 Determining Correlation Coefficients Using Regression Analysis

The experimental environment is constructed using a NISTNet WAN emulator [13], with a varying RTT and packet loss rate to determine the coefficients  $C'_1$  and  $C_2$  in Equation 2.

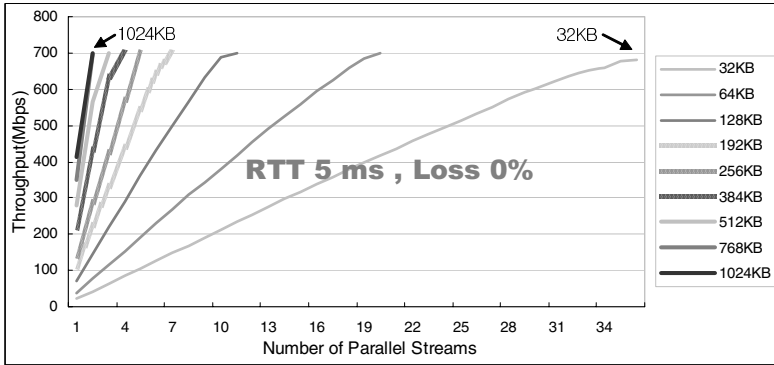
- Two 800MHz Pentium-III hosts A and B are used, and are directly connected by a 1GBps Ethernet link.
- Enable high performance options: Window Scaling, Selective Acknowledgement, Time stamps [15].
- Install the NISTNet WAN emulator on host A and impose an artificial delay.

In addition, iperf [14] is used to measure throughput. The socket buffer size can be set using the -w option. Note that if the buffer size is set to 64KB then the actual buffer size is set to 128KB. For accuracy, each experiment duration is set to 30 seconds using the -t option, and an average of 20 experiment results is used. The iperf using setsockopt(), application level system call, to set the buffer size. In this case, the buffer size is limited by the maximum system buffer size. Therefore, the maximum buffer size needs to be changed for providing enough space and buffer setting commands are summarized in Table 1.

First, the RTT is configured to 5ms by using NISTNet. In order to examine the optimal number of parallel streams, the number of parallel streams is increased until throughput is saturated. The experiment is repeated for each 32KB, 64KB, 128KB, 192KB, 256KB, 384KB, 512KB, 768KB and 1024KB socket buffer size.

**Table 1.** Maximum buffer setting for application level buffer tuning in Linux system

Sender buffer size setting	Receiver buffer size setting
<code>sysctl -w net.core.wmem_max=128388607</code>	<code>sysctl -w net.core.rmem_max=128388607</code>



**Fig. 2.** Throughput for different number of parallel streams

Figure 2 demonstrates that throughput increases whenever the number of parallel sockets or buffer size increases.

Then the optimal number of parallel streams for each buffer size is determined by following procedure. As throughput reaches its maximum, a throughput gain for increasing parallel streams is reduced. In the case of a 256KB buffer size, as the number of parallel streams is increased from 1 to 4, a steady throughput gain of approximately 148Mbps, 141Mbps, 142Mbps and 131Mbps is shown. From the point of an application using 4 parallel streams, the throughput gain decreases and the throughput does not change when using more than 4 parallel streams. To verify the result, 50 more parallel streams are provided with no additional throughput gain. Finally, it can be concluded that  $P_o$  is 5 and  $B_o$  is 256KB.

The optimal number of parallel streams for each buffer size is presented in Table 2. The result shows that  $P_o$  is in inversely proportional to  $B_o$ .

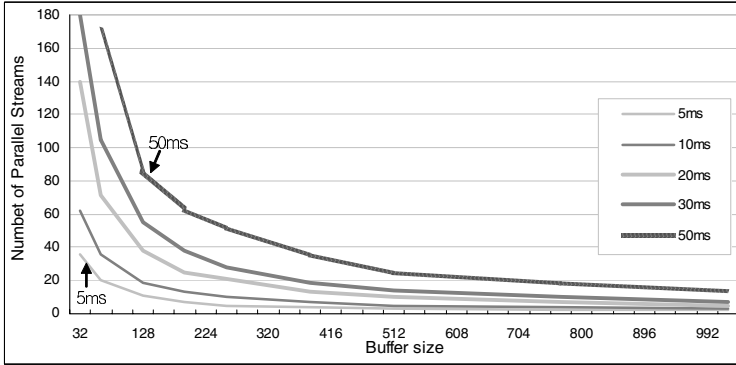
In order to examine the coefficient changes when RTT changes, the same experiments are repeated on 10ms, 20ms, 30ms and 50ms RTT, and the results obtained demonstrates a similar trend in the previous result of 5ms RTT. The maximum throughput is decreased by 10Mbps for each 10ms RTT increase.  $P_o$  for the same buffer size is increased proportional to RTT.

Among the components of equation,  $B_o$  can easily be affected by packet loss. However, research networks such as the KOREN have no packet loss. Therefore, studies considering packet loss are not referred to in this paper, and are classified as future work. Figure 3 shows the relationship between  $B_o$  and  $P_o$  for each RTT from the previous experiment result.

**Table 2.** Optimal number of parallel streams for each buffer size

	$B_o$ (KB)								
	32	64	128	192	256	384	512	768	1024
$P_o$	36	20	11	7	5	4	3	2	2



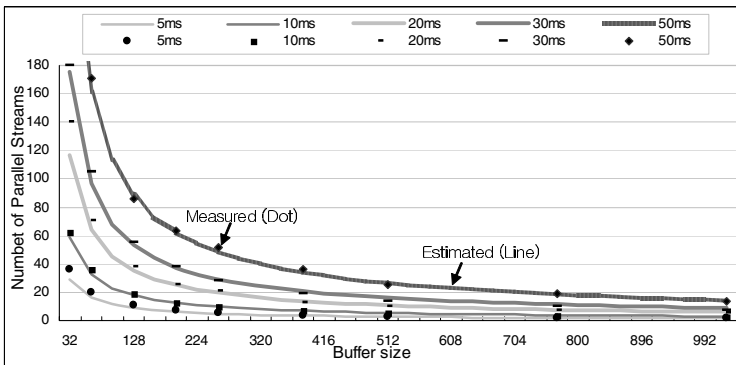


**Fig. 3.** Relationship between  $B_o$  and  $P_o$  on various RTT

From the previous result, five regression equations can be easily obtained with its own coefficients for each RTT of 5ms, 10ms, 20ms, 30ms and 50ms. Because there are many applications which provide single regression analysis functionality such as Excel and MATLAB. The regression equation for the  $i^{th}$  RTT is presented as the following form.

$$P_o = \frac{C_1''(i)}{B_o^{C_2(i)}} = C_1'(i) \times \frac{RTT}{B_o^{C_2(i)}} \tag{3}$$

Reviewing the five equations,  $C_1''(i)$  is proportional to RTT and  $C_2(i)$  is almost stationary to RTT, with minimal irregular variance. In order to present five equations as a single equation,  $C_2(i)$  must be averaged and  $C_1''(i)$  must be replaced by a constant multiplied by RTT. The constant can be represented as the average slope of  $C_1''(i)$  for varying RTT. With the additional calibration process, the representative value of  $C_1'$  and  $C_2$  is obtained as 155 and 0.86 respectively.



**Fig. 4.** Comparison between estimated and measured results

This process is called as multiple regression analysis. Finally, the equation can be written in the form of Equation 4 by replacing  $C_1''(i)$  with a constant and RTT. For convenience, RTT is used in *ms* and  $B_o$  is used in *KB*.

$$P_o \approx \frac{155 \times RTT}{B_o^{0.86}} \tag{4}$$

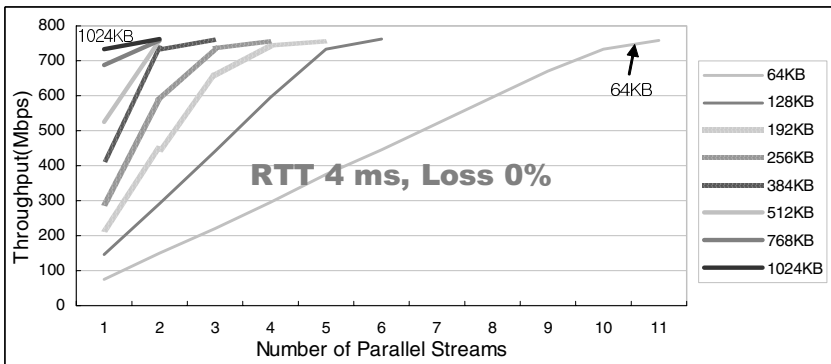
Figure 4 demonstrates the differences between measured and estimated results. The dotted graph represents the measured result and the line graph represents the estimated result. The equation measures the relationship between the optimal buffer size and the number of parallel streams with minimal 8% difference. Thus, the results of the experiment strongly support the accuracy of the proposed equation.

### 4 Verification of Relational Equation

The regression equation is developed based on a WAN Emulator. Therefore, verification of the equation on a real environment is required. A verification experiment is made on the KOREN, between Suwon and Daegu, which providing 1 Gbps bandwidth, and verifying the equation. The average RTT between Suwon

**Table 3.** The experiment environment of two organizations at a distance of 250km

KNU Server at Daegu		SKKU Server at Suwon	
CPU	AMD Opteron(tm) Processor 244 1.8GHz	Intel(R) Xeon(TM) CPU	2.40GHz
NIC	Broadcom Corp. BCM5701 Gigabit Ethernet	Intel Corp. 82540EM	Gigabit Ethernet
OS	Linux version 2.4.20-31.9smp	Linux version 2.4.20-8smp	



**Fig. 5.** Actual transfer from Suwon to Daegu

**Table 4.** The comparison of measured and estimated results

	$B_o$ (KB)							
	64	128	192	256	384	512	768	1024
$P_o$ (measured)	11	6	4	3	2	2	2	1
$P_o$ (estimated)	11	7	5	4	3	2	2	1

and Daegu is 4ms. The server specifications used for the experiment are presented in Table 3 and the results are presented in Figure 5.

Table 4 summarizes the result of Figure 5 as a relationship between  $B_o$  and  $P_o$ , comparing the result with an estimation. Based on this comparison, it can be concluded that our equation estimates the relationship between the optimal socket buffer size and the number of parallel sockets, even if it uses different hosts with differing network performance.

This result of the optimal relationship can be utilized on a decision basis for all applications using a parallel stream such as GridFTP. When these applications establish connections, the buffer size and number of parallel connection streams can be automatically configured to the most appropriate combination without wasting time based on the equation.

## 5 Conclusion

A relationship based on statistical data can be identified using multiple regression analysis. Our regression based approach starts from it. Various experiments on TCP buffer tuned parallel streams with varying RTT and packet loss rate were conducted using the NISTNet WAN Emulator to obtain single regression equation which represents a relationship of the number of parallel streams, socket buffer size, and RTT. In order to verify our approach, the equation was applied using actual data transfer between Suwon and Daegu. The verification experiment results demonstrate that the proposed scheme can predicts extremely well. The results can be utilized on a decision basis for all applications that require balancing network resources such as the socket handlers and socket buffers.

In order to improve the equation, packet loss and traffic interference must be carefully considered. Moreover, a parallel transfer management mechanism which can be applied to GridFTP has to be studied in detail using the research results.

## References

1. A. Chervenak, I. Foster, C. Kesselman, C. Salisbury, and S. Tuecke, "The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Datasets," *Journal of Network and Computer Application*, vol.23, pp. 187-200, 2001.
2. B. Allcock, I. Mandrichenko, and T. Perelmutov, "GridFTP v2 Protocol Description," *Germi National Accelerator Laboratory*, 2004.

3. D. Kakadia, "Understanding Tuning TCP," Sun BluePrints, Mar. 2004.
4. B. Tierney, "TCP Tuning Guide for Distributed Applications on WAN," In USENIX&SAGE Login, <http://www.didc.lbl.gov/tcp-wan.html>, Feb. 2001.
5. J. Semke, J. Mahdavi, and M. Mathis, "Automatic TCP Buffer Tuning," ACM SIGCOMM 1998, vol. 28, no. 4, 1998.
6. H. Sivakumar, S. Bailey, and R. L. Grossman, "PSockets: The Case for Application-level Network Striping for Data Intensive Applications using High Speed Wide Area Networks," IEEE/ACM SC2000, Nov. 2000.
7. R. L. Grossman, H. Sivakumar, and S. Bailey, "PSockets: The case for application-level network striping for data intensive applications using high speed wide area networks," Supercomputing, IEEE and ACM, 2000.
8. B. L. Tierney, D. Gunter, J. Lee, and M. Stoufer, "Enabling Network-Aware Applications," IEEE-HPDC, 2001.
9. M. K. Gardner, W. Feng, and M. Fisk, "Dynamic Right-Sizing in FTP (drs-FTP): Enhancing Grid Performance in User-Space," IEEE Symposium on High-Performance Distributed Computing (HPDC-11/2002), Edinburgh, Scotland, July 2002. LA-UR 02-2799.
10. E. Weigle and W. Feng, "Dynamic Right-Sizing:A Simulation Study," IEEE ICCN, 2001.
11. E. Weigle and W. Feng, "A Comparison of TCP Automatic Tuning Techniques for Distributed Computing," HPDC-11, 2002.
12. M. Mathis, J. Semke, J. Mahdavi, and T. Ott, "The Macroscopic Behavior of the Congestion Avoidance Algorithm," Computer Communications Review, vol. 27, number 3, July 1997.
13. M. Carson and D. Santay, "NIST Net: A Linux-based Network Emulation Tool," Computer Communication Review, June 2003.
14. M. Gates, A. Tirumala, J. Dugan, and K. Gibbs, Iperf, <http://dast.nlanr.net/Projects/Iperf/>, NLANR, 2003
15. TCP Extensions for High Performance, RFC1323.
16. Linux man page, <http://www.die.net/doc/linux/man/man7/tcp.7.html>

# Multi-level Service Differentiation Scheme for the IEEE 802.15.4 Sensor Networks\*

Euijik Kim<sup>1</sup>, Meejoung Kim<sup>2</sup>, Sungkwan Youm<sup>1</sup>, Seokhoon Choi<sup>1</sup>,  
and Chul-Hee Kang<sup>1</sup>

<sup>1</sup> Department of Electronics Engineering, Korea University,  
1, 5-ga, Anam-dong, Sungbuk-gu, Seoul 136-701, Korea

{ejkim, skyoum, shchoi, chkang}@widcomm.korea.ac.kr

<sup>2</sup> Research Institute for Information and Communication Technology, Korea University,  
1, 5-ga, Anam-dong, Sungbuk-gu, Seoul 136-701, Korea  
meejkim@korea.ac.kr

**Abstract.** In the sensor networks, the data packets transmitted by the different devices in the home networking and the industrial application maintain the different levels of importance. In this paper, we propose two mechanisms for the IEEE 802.15.4 sensor networks to provide the multi-level differentiated services which are required by each and every device. The mathematical model based on the discrete-time Markov chain is presented and is analyzed to measure the performances of the proposed mechanisms. The numerical results show the effect of the variation of the contention window size and the backoff exponent for the service differentiation on the 802.15.4 sensor networks. From the results, we derive that the contention window size is more affective than the backoff exponent on the service differentiation while the backoff exponent is more affective than the contention window size on the average delay of every device. Simulation results are also given to verify the accuracy of the numerical model.

## 1 Introduction

For the last few years, the researches on the wireless sensor networks have been increased significantly. Terms such as pervasive computing and smart spaces are being used to describe the future computing and communications. These concepts are adopted to our personal and business domains being densely populated with miniature sensors, which are constantly monitoring the environment and reporting the data to each other or to some central base station. The sensor networks cover from small applications such as the health monitoring to large applications such as the environment surveillance. In other words, it can be used widely in practical applications from the home networking to the industrial application.

The recent IEEE 802.15.4 standard for the low rate wireless personal area networks is considered as one of the technology candidates for the wireless sensor networks,

---

\* This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment).

since it supports small, cheap, energy-efficient devices operating on battery power that require little infrastructure to operate, or none at all [1, 2].

IEEE 802.11e is the research for the service differentiation. It supports the service differentiation by applying a number of queues in one node. The contention-based channel access function of IEEE 802.11e is EDCF, in which multiple queues (up to 8) are used for different priorities (up to 8). Priorities are achieved by differentiating the arbitration inter-frame space and the initial window size. However, it is not possible to use multiple queues in the sensor networks. In addition, it does not consider the high capacity traffics, which is not suitable for the sensor networks.

In this paper, we observe the data packets that are transmitted by the different devices which have different levels of importance in the sensor networks using in the home networking and the industrial application [3]. Considering the home networking, for the sensor devices performing the gate security system and the temperature sensor of the fire-warning facilities, the data packet losses for the sensor devices could be fatal. On the contrary, the data packet losses for the sensor devices which belong to TVs or refrigerators might be negligible. In addition, considering the sensor devices belong to the production line in the industrial application, each device requires the different bandwidth in a network. If there are two lines producing 30 and 60 products per second, each device in each line requires the different processing rates and the bandwidths in a network. Obviously, devices which require the high bandwidths and produce the emergency data have to take the priorities prior to other devices.

In this paper, we propose two mechanisms for the modified 802.15.4 sensor network which provide multi-level differentiated services for each and every device [4, 5]. The proposed mechanisms make it possible to provide the superior services to the device which requests the high Quality of Service (QoS) prior to other devices. We present the mathematical model for the proposed mechanisms based on the 802.15.4 sensor networks, which is based on previous works of analyzing IEEE 802.11 [6]. We consider the beacon-enabled mode with the slotted CSMA-CA algorithm in our model and assume the saturation conditions, i.e. each and every device always has a packet waiting to be transmitted, for the performance analysis. The mathematical model is based on the discrete-time Markov chain in which each component of state is representing the situation of the head packet in the queue of a device. By analyzing the Markov chain, we obtain the access probability for the device and the probability that the medium is idle. Moreover, we obtain the saturation throughput and the saturation delay. We expect the model proposed in this paper can be a reference model for the people who produce and use the sensor devices in matters of the parameter setting.

The rest of the paper is organized as follows. In section 2, we propose the operating mechanism and the mathematical model for the proposed mechanisms is presented in section 3. In section 4, we evaluate the performances of the proposed mechanisms such as throughput and delay through the mathematical analysis. Section 5 provides the numerical and simulation results which show the accuracy of the proposed mechanisms. Finally, section 6 concludes the paper.

## 2 Multi-level Service Differentiation Scheme

In this section, we propose two mechanisms, differentiation by the contention window size and differentiation by the backoff exponent, to provide the service differentiation

by multiple different level priorities in the IEEE 802.15.4 wireless sensor networks. Priority of a device is chosen through the proposed two service differentiation mechanisms and all devices are divided into multiple different priority classes. Even though it can be also considered the combined mechanism of the proposed mechanisms, we describe the mechanisms separately in this paper. Before describing the mechanisms in detail, we first state our assumptions. The 802.15.4 sensor network model that we are considering operates in beacon-enable mode with slotted CSMA-CA algorithm. In this paper we only consider the contention access period (CAP) and analyze the proposed mechanisms in the saturation mode. When the transmitted packet meets the collision, the packet is dropped and the device tries to transmit the new packet in the head of the queue. In the following we describe the proposed mechanisms in detail.

## 2.1 Service Differentiation by Contention Window Size

All devices within the network are divided into multiple different priority classes by differentiating the contention window size in phase. A CW value is set to  $CW[q]$  according to the priority of the device. Namely, the early stage of network forming phase, every sensor device is assigned with the different priority according to the importance of the function by differentiating the CW value. For example, if three priority classes, high, middle, and low, are considered,  $q$  may take the values of 0, 1, or 2. If  $q=0$ , then  $CW=CW[0]$  and it means the priority 0 class (high class), while for  $q=1$  and 2,  $CW=CW[1]$  and  $CW=CW[2]$  mean the priority 1 class (middle class) and the priority 2 class (low class) respectively. To guarantee the priority, the relation among the three values has to be as follows.:  $CW[0] < CW[1] < CW[2]$  Such a setting of the differentiated CW value within the 802.15.4 plays a similar role in AIFS differentiation depending on traffics within the 802.11e EDCF.

## 2.2 Service Differentiation by Backoff Exponent

In the early stage of the network forming phase, every sensor device is assigned with the different priority according to the functional importance by differentiating the backoff exponent. A BE value is set to  $BE[q]$  according to the functional importance of the device. For example, if three priority classes, high, middle, and low, are considered,  $q$  may take the values of 0, 1, or 2. If  $q=0$ , then  $BE=BE[0]$  and it means the priority 0 class (high class), while for  $q=1$  and 2,  $BE=BE[1]$  and  $BE=BE[2]$  mean the priority 1 class (middle class) and the priority 2 class (low class) respectively. To guarantee the priority, the relation among the three values has to be as follows:  $BE[0] < BE[1] < BE[2]$

The BE differentiation of the proposed mechanism differentiates the randomly chosen backoff duration. There is a difference between the backoff duration of the proposed mechanism and the backoff duration of the conventional 802.15.4 as follows.: At the beginning of the backoff stage 0, it chooses a random value in the range of  $[0, W_0-1]$ ,  $W_0 = 2^{BE[q]}$ , as a backoff counter. When it senses the channel becomes busy, the backoff stage of it is increased by one and the BE value is also increased by one, and the backoff counter is randomly chosen in  $[W_0, W_1-1]$  ( $W_1=2W_0$ ). More details are given to the next section.

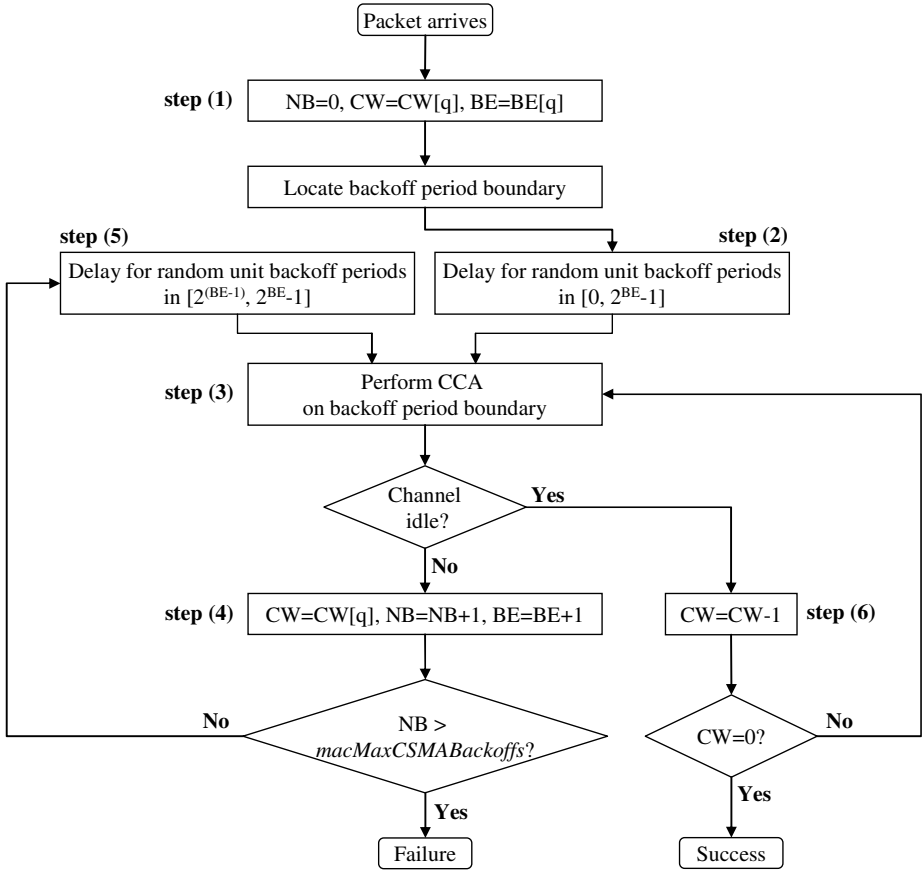


Fig. 1. The operation of the multi-level service differentiation scheme

### 2.3 Operation of Multi-level Service Differentiation Scheme

Fig. 1 describes the operation of the proposed scheme. In the figure, step (1) initialize the NB value as 0 and set the values of CW and BE according to the priority classes (one of the two or both of mechanisms can be selected depending on the requested service differentiation). In step (2), the algorithm attempts to avoid collisions by selecting a random waiting time in the range of  $[0, 2^{BE} - 1]$ . The MAC sublayer will proceed with step (3) and performs the first CCA (Clear Channel Assessment) to see whether the channel is idle. If the channel is busy, the values of NB and BE are increased by one and the CW value is set to  $CW[q]$  again which is the same value as in the first backoff stage. If the number of retries in step (4) is less than or equal to *macMaxCSMABackoffs* (the default value of which is 5), the algorithm goes to step (5). In step (5), while the backoff period within the standard 802.15.4 can take any value in the range of  $[0, 2^{BE} - 1]$ , the proposed scheme can take the value in the range excluding the last backoff stage. In other words, the range of the backoff counter is



given by  $[2^{BE-1}, 2^{BE} - 1]$ , which is the randomly selected backoff period. Such a mechanism ensures the service of the high priority class devices prior to others by increasing the differences of the backoff period. If the values of NB is above *macMaxCSMABackoffs*, the algorithm terminates with a channel access failure status. The failure will be reported to the higher protocol layers. On the other hand, if the channel is idle, the procedure goes to step (6). In other words, the CW value is decreased by one and the channel is assessed again. When the channel is idle in subsequent slots and so the CW value becomes 0, the packet transmission begins, provided the remaining number of backoff periods in the current superframe suffices to handle both the packet and the subsequent acknowledgment. Whether the transmission succeeds or fails, the algorithm goes to the next packet of the queue of the device.

### 3 Analytical Model

To analyze the proposed scheme, we introduce the following three random variables for a given device in the priority  $q$  class. Let  $n(q, t)$  and  $c(q, t)$  be the stochastic processes representing the values of NB and CW, respectively at time  $t$ . Let  $b(q, t)$  be the stochastic process representing the value of the backoff counter. Note that NB represents the backoff stage within the range of  $[0, m+1]$ ,  $m = \text{macMacCSMABackoffs}$ , and  $q$  gives the multiple priorities taking values in  $[0, \infty)$ .

The process  $\{n(q, t), c(q, t), b(q, t)\}$  forms a multi-dimensional Markov process defining the state of the packet at the backoff unit boundaries. Since every and each device has its own priority which does not change, the process  $\{n(q, t), c(q, t), b(q, t)\}$  can be written simply as  $\{n(t), c(t), b(t)\}$ . Then the corresponding state space is denoted as follows:

$$\Omega = \{(n(t), c(t), b(t)) \mid 0 \leq n(t) \leq m+1, 0 \leq c(t) \leq q, 0 \leq b(t) \leq W_i - 1, i = 0, \dots, m\}$$

where  $W_0 = 2^{BE[q]}$  and  $W_i = 2^i W_0$ .

The state transition diagram of these states is illustrated in Fig. 2. For simplicity of the notations, we use the transition probabilities  $P(i, j, k-1 \mid i, j, k)$  instead of  $P(n(t+1) = i, c(t+1) = j, b(t+1) = k-1 \mid n(t) = i, c(t) = j, b(t) = k)$ .

For analysis, we consider a fixed number of devices  $n$  with each of them is always having a packet available for transmission, i.e. saturation mode. We assume that every device is classified into 3 priority classes according to its priority consisting of  $n_0, n_1$ , and  $n_2$  devices satisfying  $n_0 + n_1 + n_2 = n$ . Then the one-step transition probabilities are given as follows.:

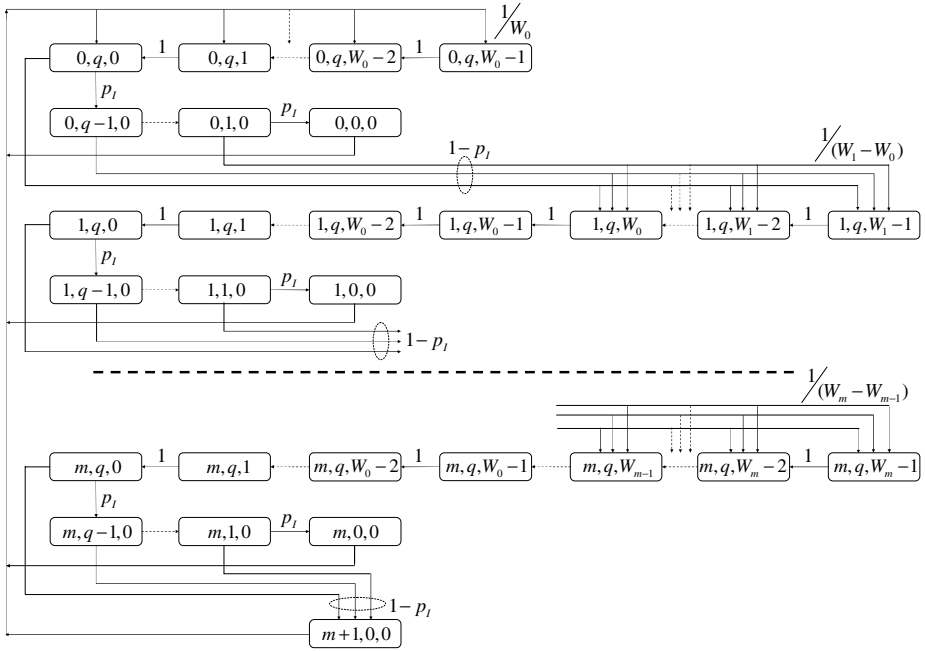
$$P(0, q, k \mid i, 0, 0) = 1/W_0, \quad \text{for } i \in [0, m], k \in [0, W_0 - 1]$$

$$P(0, q, k \mid m+1, 0, 0) = 1/W_0, \quad \text{for } k \in [0, W_0 - 1]$$

$$P(i, q, k-1 \mid i, q, k) = 1, \quad \text{for } i \in [0, m], k \in [1, W_i - 1]$$

$$P(i, j-1, 0 \mid i, j, 0) = p_i, \quad \text{for } i \in [0, m], j \in [1, q]$$

$$P(i+1, q, W_{i-1} + k \mid i, j, 0) = (1 - p_i)/W_{i-1}, \quad \text{for } i \in [0, m], j \in [1, q], k \in [0, W_{i-1} - 1]$$



**Fig. 2.** Markov chain model

Note that the states are positive recurrence and the system is stable. Therefore, there exist the stationary probabilities of the discrete-time Markov chain which is defined by  $b_{i,j,k} = \lim_{t \rightarrow \infty} P(n(t) = i, c(t) = j, b(t) = k)$ , for  $i \in [0, m + 1]$ ,  $j \in [0, q]$ , and  $k \in [0, 2^{BE} - 1]$ . Let  $\mathbf{b}$  be the stationary vector, i.e.,

$$\mathbf{b} = (b_{0,0,0}, b_{0,1,0}, \dots, b_{0,q,W_0-1}, \dots, b_{m,0,0}, \dots, b_{m,q,W_m-1}, b_{m+1,0,0}).$$

Then it satisfies the following two equations:

$$\mathbf{bP} = \mathbf{b} \text{ and } \mathbf{be} = 1, \tag{1}$$

where  $\mathbf{e}$  is the column vector whose components are consisted with 1 and  $\mathbf{P}$  is the transition probability matrix when  $\Omega$  is ordered lexicographically. By using the first part of Eq. (1), we obtain the following relations between stationary probabilities.:

$$\begin{aligned} b_{i,0,0} &= b_{0,0,0}(1 - p_i^q)^i, \quad i \in [0, m] \\ b_{i,q,k} &= b_{0,0,0}(1 - p_i^q)^{i+1} / p_i^q, \quad i \in [1, m]; k \in [0, W_{i-1} - 1] \\ b_{i,q,W_{i-1}+k} &= b_{0,0,0}\{(W_{i-1} - k)(1 - p_i^q)^{i+1} / W_{i-1}p_i^q\}, \quad i \in [1, m]; k \in [0, W_{i-1} - 1] \\ b_{0,q,k} &= b_{0,0,0}\{(W_0 - k) / W_0p_i^q\}, \quad k \in [0, W_0 - 1] \\ b_{i,j,0} &= b_{0,0,0}p_i^j(1 - p_i^q)^i, \quad i \in [0, m]; j \in [1, q - 1] \\ b_{m+1,0,0} &= b_{0,0,0}(1 - p_i^q)^{m+1} / p_i^q \end{aligned} \tag{2}$$

We obtain  $b_{0,0,0}$  by substituting Eq. (2) into the second part of Eq. (1). By substituting  $b_{0,0,0}$  into each equation in Eq. (2), we can obtain the stationary probabilities  $b_{i,j,k}$ . Then,  $b_{i,0,0}$  is derived as follows.:

$$b_{i,0,0} = \frac{2p_l^q(1-p_l^q)^i}{3-2(1-p_l^q)^{m+1} + \frac{3W_0(1-p_l^q)^2(1-2^m(1-p_l^q)^m)}{2p_l^q-1} + \frac{2(1-p_l^q)^{m+1}(p_l^q-2p_l^q+1)+2p_l^q(1-p_l^{q-1})}{1-p_l}}$$

With these stationary probabilities, we find the probability that the device transmits a packet at the boundary of a backoff period which will be denoted by  $\tau$ . Let  $\tau_q$  be the probability that a device in the priority  $q$  class transmits during a generic slot time. Then we have

$$\tau_q = \sum_{i=0}^m b_{i,0,0}. \quad (3)$$

Since the device belongs to the different priority classes has a different CW[q] value or BE[q] value, the index  $q$  of  $\tau$  is needed to differentiate the priority classes. Therefore, during the CCA procedure, the probability that the channel is idle is given by

$$p_I = (1-\tau_0)^{n_0} (1-\tau_1)^{n_1} (1-\tau_2)^{n_2}, \quad n = n_0 + n_1 + n_2. \quad (4)$$

By substituting Eq. (4) to Eq. (3), we can solve unknown parameters,  $\tau_0$ ,  $\tau_1$ , and  $\tau_2$ , numerically and then calculate  $p_I$  from Eq. (4).

## 4 Performance Analysis

### 4.1 Throughput

Let  $p_{S,q}$  denote the probability that a successful transmission occurs in a time slot for the priority  $q$  class. Then these probabilities are calculated as follows.:

$$p_{S,q} = n_q \tau_q (1-\tau_q)^{n_q-1} \prod_{h=0, h \neq q}^2 (1-\tau_h)^{n_h} = \frac{n_q \tau_q}{1-\tau_q} p_I, \quad \text{for } q = 0, 1, 2$$

Let  $p_B$  be the probability that the channel is sensed busy in a time slot. Then it is given by

$$p_B = 1 - p_I = 1 - (1-\tau_0)^{n_0} (1-\tau_1)^{n_1} (1-\tau_2)^{n_2}.$$

Therefore, the probability that a collision occurs in a time slot for the priority  $q$  class which is denoted by  $p_{C,q}$  is given by

$$p_{C,q} = 1 - (1-\tau_q)^{n_q-1} \prod_{l \in Q, l \neq q} (1-\tau_l)^{n_l}, \quad \text{for } q = 0, 1, 2.$$

Let  $S_q$  be the normalized throughput for the priority  $q$  class. Let  $\delta$ ,  $L$ ,  $T_{S,q}$ , and  $T_{C,q}$  denote the duration of an empty time slot, the payload size, the average time that the channel is sensed busy because of a successful transmission for the priority  $q$  class, and the average time that the channel has a collision for the priority  $q$  class, respectively. Now we can express the normalized throughput  $S_q$  as the following ratio:

$$S_q = \frac{E(\text{payload information transmitted in a slot time for the priority } q \text{ class})}{E(\text{length of a slot time})}$$

$$= \frac{p_{S,q}E(L)}{p_I\delta + p_{S,q}T_{S,q} + (p_B - p_{S,q})T_{C,q}}$$

where  $T_H$ ,  $T_{E(L)}$ ,  $t_{ACK}$ ,  $SIFS$ ,  $L^*$ ,  $T_{E(L^*)}$ , and  $\gamma$  denote the time to transmit the header (including MAC header, PHY header), the time to transmit the payload, the time to transmit the ACK, the time of SIFS, the length of the frame in a collision, the time to transmit a payload with length  $E(L^*)$ , and the time of the propagation delay, respectively. Note that  $T_{S,q}$  and  $T_{C,q}$  are given by

$$T_{S,q} = T_H + T_{E(L)} + 2SIFS + \gamma + t_{ACK}$$

and

$$T_{C,q} = T_H + T_{E(L^*)} + 2SIFS + \gamma + t_{ACK}.$$

### 4.2 Average Delay

In this paper, the delay of a packet is defined as the time elapsed from the instant of the generation of the packet to the instant of the successful reception or drop of it. Let  $D$  be the time to process the transmission. Note that  $D$  is a random variable depending on the priority  $q$ . Let  $E(D)$  be the mean value of  $D$ . In the discrete-time Markov chain model of section 4, for each state  $(i, j, k)$ , the mean value of the delay  $E(D_{i,j,k})$  can be presented as follows:

$$E(D_{i,0,0}) = \delta, \quad \text{for } i \in [0, m + 1]$$

$$E(D_{i,q,k}) = \delta k + \delta E(D_{i,q,0}), \quad \text{for } i \in [0, m]; k \in [0, W_i - 1]$$

$$E(D_{i,j,0}) = \delta p_I (1 + E(D_{i,j-1,0})) + \delta \sum_{k=1}^{W_i} \frac{1 - p_I}{W_i} (1 + E(D_{i+1,q,W_{i+1}-k})),$$

$$\text{for } i \in [0, m - 1]; j \in [1, q]$$

$$E(D_{m,j,0}) = \delta \sum_{r=1}^j (p_I^{j-r} + p_I^{j-2}), \quad \text{for } j \in [2, q].$$

Therefore, the mean of total delay can be found by the following equation.

$$E(D_{total}) = \delta \sum_{i=0}^m \sum_{j=0}^q \sum_{k=0}^{W_i-1} E(D_{i,j,k}) b_{i,j,k} + \delta E(D_{m+1,0,0}) b_{m+1,0,0}$$

## 5 Numerical and Simulation Results

In this section we present the performance of the analysis and compare the numerical and simulation results to verify the accuracy of the proposed numerical model. The numerical results show the effect of the variations of  $CW[q]$  and  $BE[q]$  for the service differentiation. Simulations are performed with a simple Matlab simulator. The parameters used in the numerical analysis and the simulation refer to the BPSK mode and are listed in Table 1. Also some assumptions are made for simplifying the simulation and the numerical analysis without losing the comprehensive analysis of the model. These assumptions can be accepted in the saturation mode we consider in this paper. We assume that the size of packets is constant. In addition, we assume that the required time to transmit a packet whether it succeeds or fails is the total time to transmit a data and an ACK packet with 2 SIFS intervals so that ACK packets are never lost.

In order to verify the accuracy of the proposed model, the comparison of throughputs with a varying number of devices within the each class is presented in Table 2. The value of  $CW[q]$  is set by 2 for all the devices in every class and the values of  $BE[0]$ ,  $BE[1]$ , and  $BE[2]$  at the device of each class are set by 2, 3, and 4, respectively. As shown in Table 2, the results of simulation are almost the same as those of numerical analysis. All simulation results in the table are obtained with a 98% Confidential Rate. In the tables, the Confidential Rate ( $CR$ ) between numerical and simulation results are given, which is calculated by the following equation.:

$$CR = \left( 1 - \left| \frac{E[S_{num} - S_{sim}]}{E[S_{num}]} \right| \right) \times 100\%$$

Moreover, both results of the numerical analysis and the simulation show the throughput at a class is almost twice as that at the class of one level lower priority. Since the difference between the numerical and simulation results is negligible, in the remained figures we present the numerical results only.

**Table 1.** The parameter set used in the numerical analysis and simulation

Packet payload	816 bits	Channel bit rate	20 Kbit/s
ACK	40 bits	SIFS	12 symbols
MAC header	200 bits	Retry limit	5
PHY header	48 bits	Slot time	20 symbols

**Table 2.** Comparison of throughputs with varying number of devices

The number of node for each class			Analysis			Simulation		
High	Middle	Low	High	Middle	Low	High	Middle	Low
5	5	5	3,621.8	1,782.0	883.96	3,645.6	1,781.2	881.36
10	10	10	2,514.4	1,236.2	612.95	2,522.4	1,213.7	612.45
15	15	15	1,971.6	971.3	482.07	1,977.5	972.5	482.47
20	20	20	1,653.7	816.1	405.39	1,654.8	817.3	401.23
25	25	25	1,420	701.8	348.87	1,423	702.6	348.47

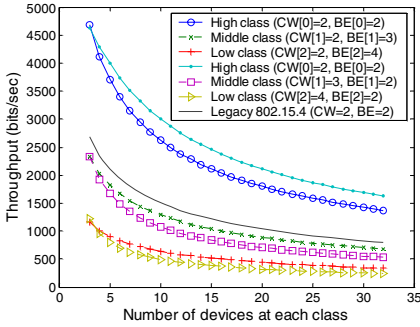


Fig. 3. Comparison of throughput

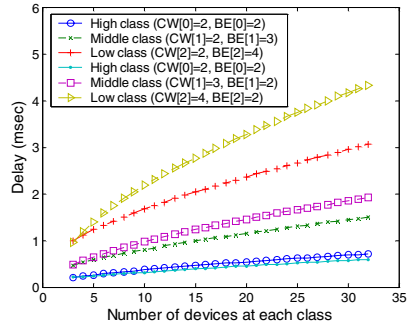


Fig. 4. Comparison of delay

The saturation throughputs for the varying values of  $CW[q]$  and  $BE[q]$  are shown in Fig. 3. In the figure, x and y axis denote the number of devices at each class and the throughput, respectively. We assume the number of devices in each class is all the same in the figures of numerical results. In other words, if x is 5, there is the total number of 15 devices. We analyze the effect of the variation of parameters  $CW[q]$  and  $BE[q]$  for a varying number of devices. The throughput of the legacy 802.15.4 is equal to the average throughput of the three classes. The throughput of high classes at the variation of  $CW[q]$  or  $BE[q]$  outperforms than that of the legacy 802.15.4 while the throughput of middle and low classes is degraded below that of the legacy 802.15.4. The throughput of a device in each class decreases as the total number of devices increases which dues to the fact that the device has to run the backoff algorithm more since more devices will lead more collision. The throughput of high class from results set with the variation of  $CW[q]$  is higher than that from results set with the variation of  $BE[q]$ . The increment of  $CW[q]$  means that the devices of lower class yield the opportunity to occupy the channel to the devices in higher class. Therefore, the probability of retrying the transmissions of a device in lower class increases as the value of  $CW[q]$  at the class increases. The delicate tuning of the throughput could be performed by varying the value of  $BE[q]$  while the value of  $CW[q]$  could be adjusted to increase the throughput of the high class.

Fig. 4 shows that the saturation delay increases almost linearly as the number of devices increases. The delay of the low class is larger than those of any other classes with respect to the variation of  $CW[q]$  and  $BE[q]$ . The characteristic of the saturation delay is that there are differences between the delay summations of all classes with the variation of  $CW[q]$  and that of  $BE[q]$ . In other words, the delay summations of all the classes with the variation of  $BE[q]$  is smaller than that with the variation of  $CW[q]$ . Therefore, to minimize the saturation delay, it is better to vary  $BE[q]$  rather than  $CW[q]$ .

## 6 Conclusion

In this paper, we proposed two mechanisms for the IEEE 802.15.4 sensor networks which provide the multiple level differentiated services for each and every device. The mathematical model based on the discrete-time Markov chain is provided for analyzing the performance of the proposed mechanisms.

The comparison of numerical and simulation results is given to verify the accuracy of the numerical model. The numerical results of several performance measures are given to analyze the effect of the variation of the contention window size and the backoff exponent for service differentiation on the 802.15.4 sensor networks. Increasing the contention window size or the backoff exponent within a class means that the devices of lower class yield an opportunity to occupy the channel prior to those of higher class. Therefore, the probability to retry a transmission at lower class increases, which dues to a busy channel, as the contention window size or the backoff exponent at the class increases. The numerical results show that the variation of the contention window size has more effect on the service differentiation than that of the backoff exponent even though the average delay of every device is affected more by the backoff exponent. The delicate tuning of the throughput could be performed by varying the backoff exponent, while the increase of the throughput of the high class could be performed by adjusting the contention window size.

The results obtained in this paper provide a criterion for using the parameters for specific purposes. Therefore, we expect the proposed model can be a reference model for the people who produce and use the sensor devices in matters of the parameter setting.

## References

1. Standard for part 15.4, Wireless medium access control (MAC) and physical layer (PHY) specifications for low rate wireless personal area networks (WPAN), IEEE Std 802.15.4, IEEE, New York, NY. (2003)
2. Mistic, J., Shafi, S., Mistic, V.B.: The Impact of MAC Parameters on the Performance of 802.15.4 PAN, Elsevier Ad hoc Networks, Vol. 2. (2004) 351-371
3. Bhatnagar, S., Deb, B., Nath, B.: Service Differentiation in Sensor Networks, 4th International Symposium on Wireless Personal Multimedia Communications, Vol. 1. (2001) 435-441
4. Robinson, J.W., Randhawa, T.S.: Saturation Throughput Analysis of IEEE 802.11e Enhanced Distributed Coordination Function, IEEE Journal on Selected Areas in Communications, Vol. 22. (2004) 917-928
5. Xiao, Y.: Enhanced DCF of IEEE 802.11e to Support QoS, IEEE Wireless Communications and Networking Conference, Vol. 4. (2003) 1291-1296
6. Bianchi, G.: Performance Analysis of the IEEE 802.11 Distributed Coordination Function, IEEE Journal on Selected Areas in Communications, Vol. 18. (2000) 535-547

# Development of Event Manager and Its Application in Jini Environment\*

YoungLok Lee<sup>1</sup>, HyungHyo Lee<sup>2</sup>, Seungyong Lee<sup>1</sup>, InSu Kim<sup>1</sup>, and BongNam Noh<sup>1,\*\*</sup>

<sup>1</sup> Dept. of Information Security, Chonnam National University, Gwangju 500-757, Korea  
{dogu, birch, agisong, bongnam}@lsrc.jnu.ac.kr

<sup>2</sup> Div. of Information and EC, Wonkwang University, Iksan 570-749, Korea  
hlee@wonkwang.ac.kr

**Abstract.** Ubiquitous computing services have to adapt to the context information. These services have to communicate with each other through fixed network or ad-hoc, it is the ubiquitous middleware to be able to help those services. With regard to the adaptation of middleware's components, context manager and event manager are required. Recently there is a widely used middleware, Jini, but it is distributed without the event manager services. Therefore we implement the event manager which can manage events in Jini environments, and describe ubiquitous computing applications running environment using our event management system. Our event manager is implemented by modifying Javaspaces.

## 1 Introduction

An event in ubiquitous computing can be defined as an object that contains information about external status in which other software components are interested. There are various events ranged from low level signals generated by sensors to deduced valuable information in high level. Users in the ubiquitous computing environment should be able to adapt themselves to their current context information and high level information generated by these events.

Jini, the Home Network middleware, helps new service components connect to Home Networks at any time and helps clients promptly use them without extra settings, and even in the case of service component upgrade, the existing client service can operate with no problems. If there are any interested changes in the outer world, whether Jini services are in local or not, it is necessary for them to be asynchronously notified. Jini can achieve it by using the event notification concepts as other java components. Jini's lookup service finds the service which client is looking for and, if found, hands over a proxy. However if not found, the client registers themselves with the lookup service as a remote event listener so that she can be notified when the event she is looking for is written to lookup.

---

\* This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment).

\*\*Corresponding author.



Even though there is a way of using Javaspace services or lookup services in order to manage events. In the ubiquitous environments which contain a lot of sensors or context managers, we need to systematically manage events by putting distributed event managers. By modifying the JavaSpace, we develop the event manager to manage events.

This paper consists of as follows;

Chapter 2 shows related work, and chapter 3 explains the design and implementation of event management system based on JavaSpace. In chapter 4, we explain how context managers and application services can utilize our event manager. Finally, makes a conclusion and further research works.

## 2 Related Work

The CORBA Event Service[1] standardized the transfer of asynchronous notifications between objects. An object that generates events is called supplier whereas an object which receives events is called consumer. A supplier passes an event to a consumer by invoking an appropriate method on the consumer interface. Suppliers and consumers can be decoupled from each other by event channel objects. An event channel forwards all events it receives from any of its suppliers to all the consumers that have registered with the channel. The events themselves can be either of generic or typed format. The standardized CORBA Notification Service[2] developed by the OMG Telecommunications Domain Task Force provide much more flexible event filtering capabilities. It is possible to establish user-defined event filters by assigning priorities to events, generate timestamps, and introduce QoS criteria for the handling of events.

One of middlewares is the Gaia of the University Illinois under active research. The event manager of Gaia[3] satisfies many of general needs in event management. The event manager distributes load among multiple machines. It creates event channel factories remotely on pre-configured machines whenever it decides that the existing event channel factories are already overloaded. In essence, the event manager keeps state for the channels it creates and recreates them if they crash, transparent to the event consumers. Event manager service implementation of Gaia makes use of CORBA event service as the basic event storage.

The Java Message Service (JMS)[4] is a Java API that allows applications to create, send, receive and understand messages. It defines a common set of interfaces and associated semantics that allows Java programs to communicate with other messaging implementations. JMS provides a loosely coupled architecture that supports asynchronous communication and guarantees reliable delivery of messages. The specification provides for both point-to-point messaging using queues and the publisher/subscriber approach using topics as intermediaries. Messages can be consumed both synchronously ("pull") and asynchronously ("push"). It also has message filtering capabilities in the form of message selectors based on a subset of SQL92 conditional expression syntax.

Distributed events in the Jini[5] framework allow an object in one virtual machine (VM) to register interest in the occurrence of an event in another object, possibly running in another VM, and receive notification when such an event happens. The Jini Event System uses the Jini Lookup Service for naming which can be optionally used

with Java Native Directory Interface (JNDI)[6]. An event is a Java object that can be subtyped for extensibility. The listener interface is simple and aids in the use of a flexible publisher model. Jini supports leasing and uses Java-RMI as the communication substrate. It leverages the built-in security of Java. However, Jini events are designed to work only in the Java environment and are not equipped to work with firewalls and Network Address Translation (NAT).

### 3 JavaSpace-Based Event Management System in Jini

JavaSpaces™ are a networked repository for java objects. One of the functions of JavaSpace is to notify an event to the entity who is interested in the event object when that event object is stored in the JavaSpace. Therefore, notification services can be implemented using JavaSpaces. However there are some problems with the function of event managers, so we implement JS-EM(JavaSpace-based Event Management System) by modifying JavaSpace. In this chapter we briefly introduce JavaSpaces and then review the issues when we implement event managers with pure JavaSpace. After that, we explain how we implement our event manager.

#### 3.1 JavaSpaces

JavaSpaces[7] are a Jini service that allows clients to share objects. Its goal is to facilitate cooperative distributed computing. It is a reliable distributed storage system. JavaSpaces should be thought of as a “place” in which it is possible to store and share information in the form of objects. Objects flow around space-based systems, allowing clients to share and store information and also, via the use of dynamic class loading, share behavior. In JavaSpaces, all objects must also implement the Entry interface. Entries are found in association or matching.

JavaSpaces use templates to match entries within the Space by their field’s values and type. JavaSpaces do use the value of the objects to do the matching. JavaSpaces export proxies to clients dynamically, via the Jini lookup service, so they are naturally distributed. JavaSpaces exist outside of other applications or services because they are themselves, but applications and services that use them import parts of the space to interact locally with the JavaSpaces interface.

#### 3.2 The Problems of Implementing Event Manager Using JavaSpaces

Event managers can be implemented with JavaSpaces. Figure 1 shows the expected procedures of the event management when the event manager is implemented using JavaSpaces. There are, however, some issues with this implementation.

If more than one event producers write the same type of objects on JavaSpace and then event consumers are trying to take a object with a read() method, there is a problem that only firstly matched object will be returned. For example, let us assume that every 4 seconds, 4 events with the same type are written on JavaSpace, and their lease time is 10 seconds. In this case, as we can see in figure 2, from the second event object forth, there is the difference in event objects which event producers issued and event consumers read.

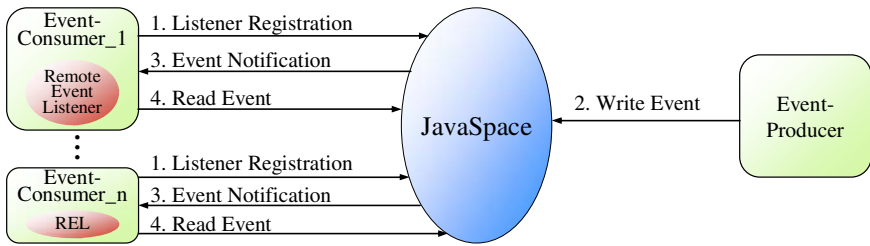


Fig. 1. Expected procedures of the event manager with JavaSpaces

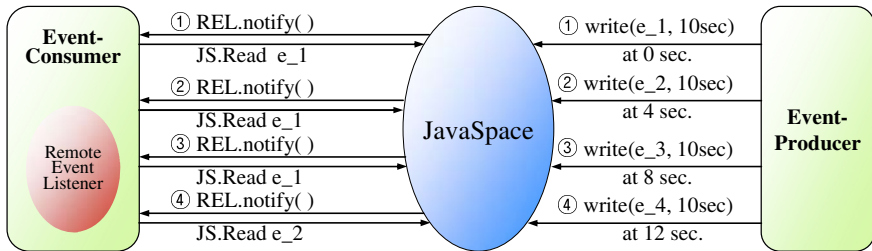


Fig. 2. Events taken by consumer with JavaSpace implementation

### 3.3 JS-EM Using the Modified JavaSpace

Our event manager (JS-EM : JavaSpace-based Event Manager) is made by modifying JavaSpace. In the existing JavaSpaces, event consumer should read the corresponding event by calling a read( ) method which is a JavaSpace API with the information of the event object which her event handler received. Granting that it can read events, there is no guarantee that it can obtain the correct one.

Hence by modifying JavaSpace for it to send the event object with the event notification as well, as in figure 3, event consumers can immediately receive events without calling a read( ) method.

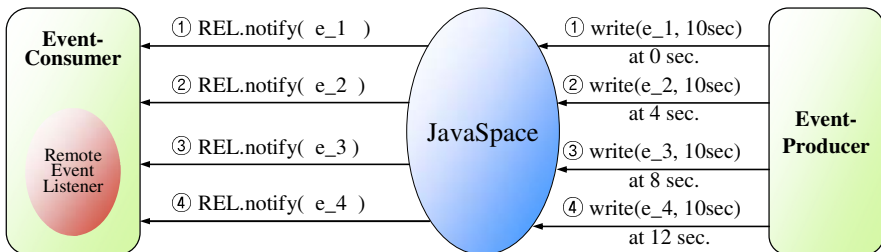


Fig. 3. Events consistency between sent events and obtained events

The procedure that the event consumer takes events generated by the event producer is as follows:

1. As JS-EM itself is registered as service in the Jini LookUp service, the event consumers or the event producers can search the JS-EM and use it. An event consumer registers herself to JS-EM as a listener of the event he or she is interested in.
2. JS-EM takes the stub of event listener for the communication with the event consumer through web server and by doing this, there accomplishes the channel between event listener and JS-EM.
3. The event producers write its events to the JS-EM.
4. Events written to the channel in JS-EM are transmitted to the event consumer via previously registered event listener.

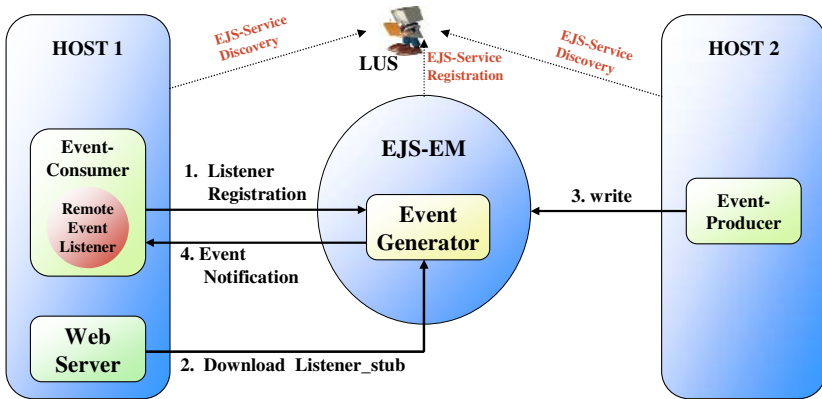


Fig. 4. Event service procedures using the extended JavaSpace

Our event manager provides a model for decoupled communication among event consumers and event producers. It allows creating and deleting channels. And through our event manager service, the context management service is able to generate the high level context information.

In a room, if a sensor sends location position information of a person after sensing, the consumer who tries to achieve the event is able to take the event through JS-EM. It means that the event is saved in the event channel generated by event manager, and then more than two event consumers, registered in the event channel as listener, are able to receive the event whenever the event producer writes the event into the event channel.

## 4 Prototype Implementation and Analysis

Our prototype is implemented in Linux/Windows OS, JDK 1.3 and Jini 1.2 development environment. We implemented our event manager by modifying JavaSpaces as described at chapter 3. In this chapter we describe how event managers having our proposed extended JavaSpace can be utilized. To implement our prototype first we

propose architecture and then make scenarios. After that, we implement the application of each scenario by using Macromedia’s flash and Sun’s Java.

### 4.1 Architecture

In figure 5 which is the overall architecture of our prototype which we want to implement, it describes how application devices and sensor devices can use our event managers. TV itself, an application device, registers at JS-EM to listen to needed events. Sensor devices write their issued events on JS-EM, and JS-EM notifies it to a context manager.

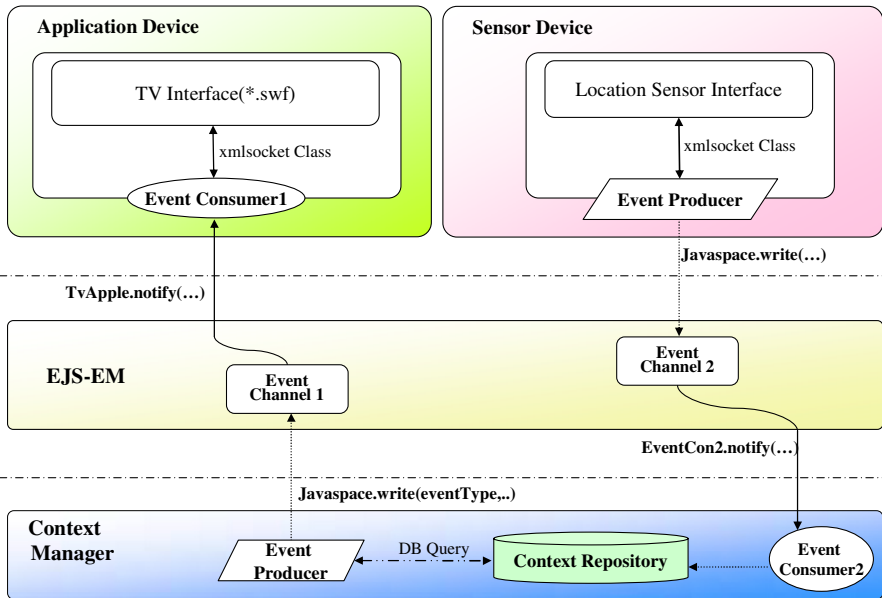


Fig. 5. Architecture of our event system

#### 4.1.1 Devices

For our demonstration we made the virtual S/W devices with Macromedia’s flash. Necessary devices for the demonstration are as follows.

- 1) Sensor devices: Transmit the event informations of location, temperature, and pressure to the context manager.
- 2) Application devices: Devices, such as a TV or a refrigerator, which adapts to events coming from context managers

#### 4.1.2 Event Manager

Event managers play roles of conveying issued events from sensors to the context manager or transmitting high-quality events collected by the context manager to application layer’s devices.

### 4.1.3 Context Manager

The context manager processes context information coming from sensors, stores it, and converts it into high layer's context information according to the inference policy. This context information is the events which application layer's devices want, so to inference those events the context manager provides GUI for setting the inference policy. As this component is beyond our paper, we omit the explanation for this.

## 4.2 Scenario

The scenario for our prototype implementation is as follows.

*6 PM, July 10, 2005. After finishing her daily work, Alice headed to home. On her way home she stopped by a fruit store. When she arrived at her home, the light of the porch automatically turned on, and she entered after authentication. First she changed her dress in her room, ate dinner, and sat down on the sofa. The context manager which recognized that Alice sat down on the sofa transmitted her favorite channel information to TV based on her preference and turned on TV. During watching TV, she recognized that she left her wallet in the fruit store. So she left her house in a hurry with TV on. TV stored the current TV status and turned off itself after receiving an event from the context manager representing Alice left. When she came back and sat on a sofa, the TV channel which she saw right before she left is on.*

### 4.3 TV On/Off Algorithm

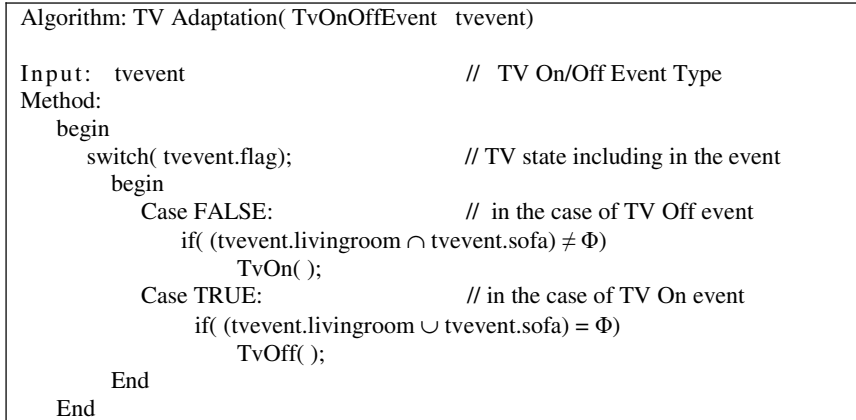
The on/off event classes which a TV device received is as follows.

```

Public class      TvOnOffEvent extends JSEvent
{
    public Boolean flag ;                // TV State(false: On, True: Off)
    public Integer counter ;             // number of the family
    public String[ ] livingRoom ;        // family in living room
    public String[ ] sofa ;              // family in sofa
    public TvOnOffEvent(Integer family, Boolean currentTvState)
    {
        totalcounter = family;           // assign the number of family
        flag = currentTvState;           // current TV On/Off State
        livingRoom = new String[counter];
        sofa = new String[ counter ] ;
    }
}

```

The adaptation algorithm of event consumer after receiving TV's on/off events is as follows.



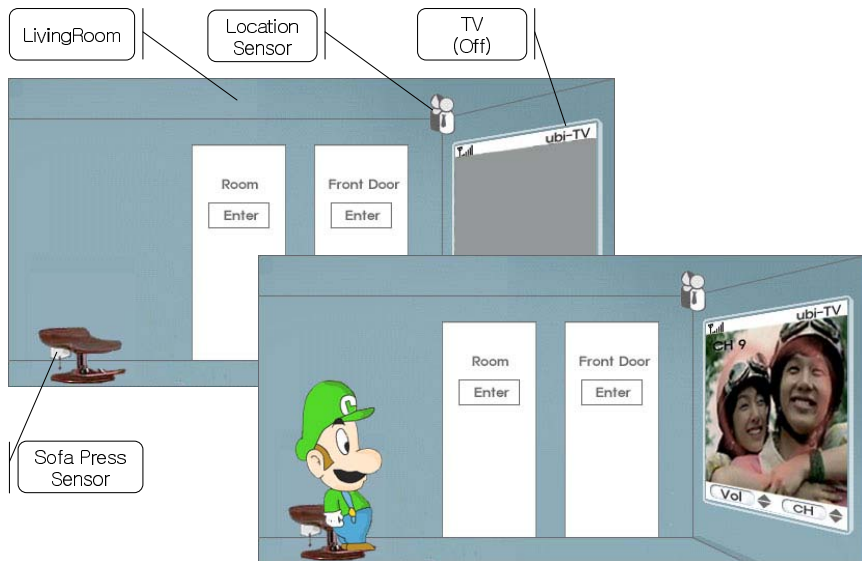
**Fig. 6.** TV On/Off Adaptation Algorithm

#### 4.4 TV On/Off Demonstration

The procedures for TV to be on or off are as follows.

1. A TV application device registers listener at JS-EM to obtain on/off event.
2. If low level events are issued from sensors, the context manager infers the high level's events from them which can make TV on or off, then notifies a TV device via an event manager.
3. The TV device adapts to on/off events

Below figure 7 shows our prototype implementing the previous scenario.



**Fig. 7.** Demonstration of our prototype implementing our scenario

### 4.5 System Analysis

As our event manager, JS-EM, has whole advantages of JavaSpace, it is cooperative and scales well for loosely-coupled systems. In our JS-EM, if the event producer write events on JS-EM with a write( ) method, then it conveys an event object to the consumer through the parameter of a notify( ) method. When we compare JavaSpace with our JS-EM from the view of event management the result is like Table 1.

**Table 1.** The comparison of JavaSpaces and JS-EM

		JavaSpaces	Our JS-EM
Num of call	Listener registration	5	5
	write( event )	1	1
	Listener match module	1	1
	notify( )	5	5
	Read( )	5	0
	Object match module	5	0
Characteristics		Event consumers cannot make sure that they are taking the event notified by JavaSpace.	Event consumers can make sure that they are taking the event notified by JS-EM

In table 1, we assume that there are five clients waiting for one same type event. This table shows that our JavaSpace much faster and gives less load because it does not call read( ) and object match modules. In the previous JavaSpace, as it returns the firstly matched object when calling a read( ) method, there is no guarantee that the very event object which issued the event returns. On the other hand, in the modified JavaSpace, as it returns the object which issued the event when notifying the event consumer, the object which issued the event can be correctly transmitted.

## 5 Conclusions and Further Work

In the ubiquitous environments, above all adaptation is the most important. For this, we made the event manager so that we can manage events in distributed environments by modifying JavaSpace. As our event manager, JS-EM, inherits the properties of JavaSpace, it can have whole advantages of JavaSpace as it is.

In the future work, we want to control events so that only owner who has the right authority can take events by extending our Jini. We will clarify differences between context manager and event manager in our further research. Formalization of ACLs is important because it possibly moves with events. In addition, we should consider where to put ACL related to event authorization. In our prototype, we only put it in the event producer in order to observe its possibility.



Because in order to minimize the event manager's burden, the event manager only relays the event, the trust checking on the side of the event consumer is in the right process, appropriate encoding is necessary. In a sense that computing power of the event consumer is available, the first or the third method is prominent in this paper. Within the ubiquitous environment, security entities should be assigned as new service, because client should keep working with substitute service.

## References

1. CORBA services: Common Object Services Specification, OMG Specification, Object Management Group, Nov (1997)
2. Siegel, J.: CORBA 3: Fundamentals and Programming, Object Management Group (2000)
3. B.Borthakur: Distributed and Persistend Event System For Active Spaces. In: Master Thesis in Computer Science, Urbana-Champaign: University of Illinois at Urbana-Champaign (2002)
4. Richard Monson-Haefel, David Chappel: Java Message Service, O'Reilly (2000)
5. W.Keith Edwards, W.Edwards: Core Jini, Pearson Education (2000)
6. Rosanna Lee, Scott Seligman: JNDI API Tutorial and Reference: building directory-enabled Java™ Applications(Paperback), Addison-Wesley Professional (2000)
7. Philip Bishop and Nigel Warren: JavaSpaces IN PRACTICE, Addison-Wesley (2003)

# On Scalability and Mobility Management of Hierarchical Large-Scale Ad Hoc Networks

Ming-Hui Tsai<sup>2</sup>, Tzu-Chiang Chiang<sup>1,3</sup>, and Yueh-Min Huang<sup>1</sup>

<sup>1</sup> Department of Engineering Science,  
National Cheng-Kung University, Taiwan, R.O.C  
huang@mail.ncku.edu.tw

<sup>2</sup> National Tainan First Senior High School, Taiwan, R.O.C.  
colux@tnfsh.tn.edu.tw

<sup>3</sup> Department of Information Management,  
Hisng-Kuo University of Management, Taiwan, R.O.C.  
n9892105@mail.ncku.edu.tw

**Abstract.** With the increased interest in the mobile wireless communication, the development of large-scale ad hoc networks has drawn a lot of attention and the scalability of ad hoc networks has been the subject of extensive research. The ad hoc network topology changes frequently and unpredictably, and mobility become extremely challenging in the circumstance. So, the broadcast storm becomes a very serious problem to migrate into such networks for the applications of group communications. The main concept of virtual subnet technology is the capability to group users into broadcast domains, which divides a virtual subnet into logic, instead of physical, segments and reduces the traffic overhead. With this characteristic, we propose an interoperability network model integrating self-organizing ad hoc networks and the Internet/a conventional network with the partition of physical/virtual subnets. Moreover, we describe a protocol to establish the virtual broadcast domains by using the IPv6 addressing concept in ad hoc networks and perform IP-based network communications in a multi-switch backbone. The hierarchical networks, physical/virtual subnets, addressing method and mobility management are described, and some performance issues are evaluated.

## 1 Introduction

Mobile ad hoc networks need the flexibility to collect more than two devices equipped with wireless communication and networking capability. Many handhelds, PDAs, laptops, notebooks and even mobile phones now also include wireless connectivity as a standard feature, and can access the Internet by a central system administration (like base stations in a cellular system or access-points in a wireless LAN) anytime, anywhere [4, 5, 11, 12]. However in some circumstances a group of people makes up a party, a conference, or participates a camping site, and there has been a growing interest in rapidly deployable and dynamically reconfigurable wireless network supporting multimedia communication (data, voice and video) not just voice as mobile phone does. Each node acts as a router and is responsible for

dynamically discovering other nodes which it can directly communicate with [6, 7, 8, 9]. The network topology of an ad hoc network changes frequently and unpredictably, so scalability and mobility become extremely challenging in the circumstance we mentioned previously. For many multimedia applications, however, it is desired that an inter-working functionality between the protocols in the ad hoc network and the IP-based Internet. Ad hoc networks should be adapted to deploy rapidly and can freely communicate with the traditional mobile wireless networking systems without the support of the pre-existing network infrastructures [1, 2, 3, 10].

However, the broadcast storm becomes a very serious problem for ad hoc networks to migrate into such networks for the application of group communications. The main concept of virtual subnet technology is the capability to group users into broadcast domains, which divides a network into logic, instead of physical, segments and reduces the traffic overhead. The network-layer address is used to inform the router physical segment where must send data to. While clients and servers may be distributed anywhere on a network, they are grouped together by virtual subnet technology, and broadcasts are sent within the same virtual subnet [16, 17, 18].

Previous work on mobile radio networks with heterogeneous communication and dynamically changing topology concentrated primarily on mobility and/or broadcast storm problems in arbitrary geographical topologies. To solve mobility problem and improve network performance and signaling/control overhead in such a heterogeneous scenario were proposed. More recently, the issues of cluster backbone and multi-cluster architecture for mobile ad hoc networks were also proposed [13, 14, 15, 19].

This paper addresses the interworking between ad hoc networks and Internet Protocol (IP-based) networks, where we focus on network partitioning can improve or solve critical functions such as mobility management, routing and broadcast problems. It can be observed in this type of architecture using network partitioning concept may result in larger scalability. To achieve this network interconnection using partitioning network, this heterogeneous communication establishes the installation of gateways with the help of specific access routers, which understand the both protocols of the ad hoc network and the IP suite. We also discuss an architecture based on a specific logical topology by partitioning a mobile ad hoc network into logically independent subnets.

## 1.1 Research Contributions

We consider a core network of 802.1Q-liked switches interconnected by trunk lines in ad hoc networks, and this network can span one or more small towns in sparsely populated areas, interconnecting communities as well as company LANs. Our intent is to provide both telephony and data services over the same technology (Ethernet), stack (IP), with seamless integration with the Internet. Our work chooses to distribute the bandwidth according to a hierarchical link sharing model.

The rest of this paper is organized as follows. We describe the problem for this network model in Section 2. Section 3 addresses in detail for the infrastructure of the scalable environment with ad hoc networks, Section 4 performance simulations with NS2 module are described. Finally, section 5 provides our concluding remarks and future works.

## 2 Problem Description

With the fast development of wireless technology, the ad hoc network is walking out from research papers and becoming closer to the common consumers. A large-scale network issue will bring ad hoc networks a new future for distributed communication and computing. The motivation behind our approach is that network partitioning can improve some critical functions as broadcast storm problems defined such channel access, routing, mobility management and scalability, while reducing signaling/control overhead within a hierarchical network.

Hence broadcasting in a shared wireless medium may result in multiple nodes receiving a transmission destined for a single node, and ultimately, in multiple transmission mutually interfering at a single node. Nodes can reduce the chances of interference by separating transmissions in time, space, frequency, or spreading code. By coordinating this separation instead of acting independently, nodes can further reduce the chances of interference and hence increase network throughput. The partition-based control structure provides a natural organization of network nodes that simplifies coordination of transmission among neighboring nodes by geographic area. It can be observed in this type of network that portioning may result also in lower congestion and high throughput compared to one large network.

This paper also discusses a network hierarchical architecture based on a specific logical topology superimposed over a physical topology (determined by transmission coverage of network nodes in the same geographic area); the architecture selects links to be affiliated (logical links) with physical links. Our main concern is finding an efficient logical topology and a suitable routing procedure which result in high performance and node mobility. In this architecture, network nodes of a cluster ad hoc network are grouped into two types of subnets: physical and virtual, and may dynamically change their affiliation with these subnets due to their mobility. Each node is allocated an address based on its current subnet affiliation and associated by IPv6 addressing. We consider networks that have several tens to several hundreds mobile nodes within this hierarchical network. We assume that there exists a channel access protocol which resolves contentions and/or interference in the network.

### 2.1 Network Partitioning

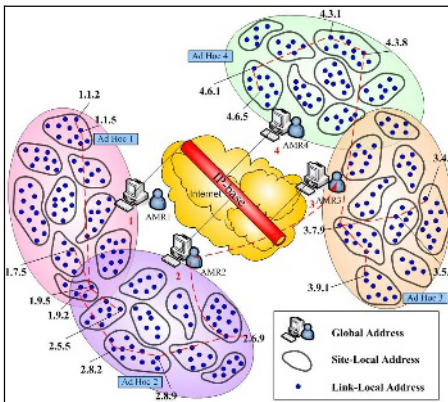
In the flat ad hoc network, all nodes participating in are assumed to be equal and share the same radio capacity. It has been proven that the flat network scheme works well in small size networks (e.g., fewer than 100 nodes) [20, 21]. However, when wireless network size increase (beyond certain thresholds), it becomes infeasible because of link and processing overhead. Building a physical hierarchical structure ad hoc network is a very promise way to solve the scalability problem. The most popular way of building hierarchy is to partition the networks into some clusters or group nodes geographically close to each other into clusters.

There are several approaches to construct the cluster; these include (1) Lowest-ID Cluster Algorithm (LID) [22], (2) Highest-Connectivity Cluster Algorithm (HCC) [23], (3) Least Clusterhead Change Algorithm (LCC) [24], (4) Distributed Clustering Algorithm (DCA) [25], and (5) Weighted Clustering Algorithm (WCA), [26]. The existing clustering algorithms differ on the criterion for the selection of their

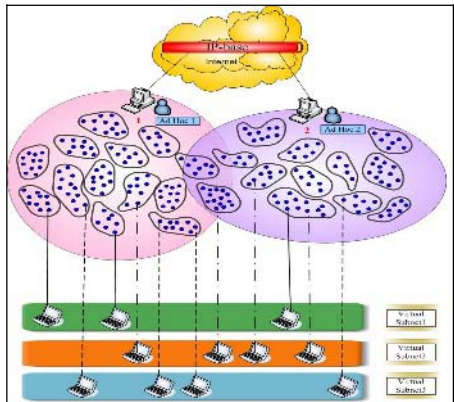
clusterheads. Once a node becomes the clusterhead, neither that node nor its members can participate in the cluster election algorithm. The election algorithm will terminate once all the nodes either become a clusterhead or a member of a clusterhead.

### 3 Architecture of a Hierarchical Large-Scale Network

As shown in Figure 1, four groups of mobile nodes form an internetworking wireless ad hoc network with IP-based Internet. The traditional ad hoc networks do not access the Internet or wired networks. However, due to the rapidly grow of the 3G communication and mobile devices, the hierarchical mobile communication architecture will be a new trend in the future. The communication between each node in this network infrastructure is established by using agent multi-route equipments and wireless multi-hop paths. By using this architecture, some mobile nodes (MNs) in this ad hoc network can access the Internet or communicate with ad hoc nodes of different wireless groups. Some critical issues for these MNs are their mobility management, the overhead and the load analysis for the agents to adapt well to dynamic topology changes by partitioning a mobile network into logically independent virtual subnets.



**Fig. 1.** An internetworking wireless ad hoc network with IP-based Internet (including Routing Path)



**Fig. 2.** An IP-based Internet with two ad hoc network groups and three virtual subnets

Hence mobile nodes may change their affiliation of physical and virtual subnets due to their mobility. First, in order to access the IP-based Internet and form the physical/virtual subnet, a set of Agent Multi-Route Routers (AMRs) is established in advance. The AMRs are connected to the Internet part as a gateway and communicate with the mobile nodes in wireless ad hoc networks via wireless transceivers. Each AMR consists of three components:

- 1.) IP-based component with traditional IP-based protocol suite installed is designed to connect with IP-based Internet (either wired or wireless);

- 2.) Ad hoc component with ad hoc related protocols installed is connected with ad hoc networks through a wireless interface;
- 3.) Virtual subnet core component provide the intelligence to make filtering and forwarding decisions with configuration tables. Virtual subnet technology allows the nodes of different ad hoc network groups connected to form a virtual subnet, as Figure 2 depicted.

### 3.1 Addressing Autoconfiguration with Physical/Virtual Subnet Concepts

We specify a mechanism by which mobile nodes in an ad hoc network may autoconfigure IPv6 addresses depending on their current physical and logical connectivity and address availability which make the IP address unique throughout the connected portion of the hierarchical network. Since Ipv6 defines both a stateful and stateless address autoconfiguration to enable plug-and-play networking of devices and reduce administration overhead. In general, each mobile node may autoconfigure a link-local address first and verifying its uniqueness on a link by using the duplicate address detection, and determine what information should be autoconfigured (addresses, other information, or both). However, in order to speed the autoconfiguration process a mobile node may generate its link-local address in parallel with waiting for a Router Advertisement to generate a site-local and/or global address.

In order to improve the scalability of our proposed network, this hierarchical network is segmented into two parts,  $x$  groups of ad hoc networks and an IP-based Internet. Assume that each group of ad hoc networks is segmented into  $y$  physical subnets each containing up to  $z$  mobile nodes. We describe the address space over an alphabet of size  $m = \max(x, y, z)$  containing the numbers  $0, 1, 2, \dots, m-1$ . Each mobile node in the cluster ad hoc networks is given a triple number, where the global significant digit (GSD) is a digit in IP-based, the site significant digit (SSD) is a digit in site-local-based and the link significant digit (LSD) is a digit in link-local-based. Therefore, the total number of mobile nodes possible is  $N = xyz$ . The partitions of these groups are the basic building blocks and the Ipv6 address autoconfiguration process of the network. The address of a mobile node is separate to two: network prefix, which identifies ad hoc group networks and physical/virtual subnet ID, which identifies the affiliations of mobile nodes. Physical/virtual subnet ID is configured by the node on its current subnet affiliation as a result of its logical-topology or logical-link position relative to other mobile nodes, and prefix is determined by the AMR.

### 3.2 Node Affiliations

Because ad hoc networks do not allow the aggregation techniques especially about scalability issue that are available to traditional IP-based routing protocols. In general, loss of aggregation results in bigger route table for proactive protocol or longer delay for reactive protocol. Our node affiliations maintain aggregation for ad hoc networks and involve in a physical subnet which nodes are close in a local geographical area and a virtual subnet which nodes form a logical network. Figure 2 also shows a hierarchical network with the aggregation concept of physical subnets and virtual subnets. We use the GSD to present a cluster ad hoc network group that all nodes in this group partition into physical subnets with the same SSD and virtual subnets with the same LSD. Therefore in this hierarchical network architecture, network nodes are grouped into

three types of clusters/ or subnets which mobile nodes may dynamically change their affiliation with these subnets according to their mobility. Members of different physical subnets are clustered together to form virtual subnets, each of which ideally spans all physical subnets and is used to provide communication paths among distant nodes.

### 3.3 Routing

Using hierarchical routing, the groups of ad hoc networks are physically and logically separated into different subnets. First, we describe a shortest path routing procedure. When a mobile node receives a packet, it checks the network prefix and the physical/virtual subnet ID. If it is the destination, it stops the relay process and operates the packet. If the mobile node is not the destination, the communication in this heterogeneous environment can be categorized into two scenarios:

**Communication among ad hoc mobile nodes across the Internet.** If a mobile node wants to communicate with the mobile nodes across the Internet, it knows from the network prefix (GSD) of the destination address that this destination belongs to different AMR. In this procedure routes traverse one digit at a time in the same physical subnet. For example, when the address of the source is 2.8.2 and the address of destination is 3.4.9, this procedure would use the routing path  $2.8.2 \rightarrow 2.8.9 \rightarrow 2.6.9 \rightarrow \text{AMR}(2) \rightarrow \text{AMR}(3) \rightarrow 3.7.9 \rightarrow 3.5.9 \rightarrow 3.4.9$ . Once the AMR receives the forwarding packets, it sends them to the AMR of the destination across the Internet.

**Communication among ad hoc mobile nodes in overlay ad hoc groups.** This routing is performed in two phases. In the first phase routing is performed as the previous scenarios, Communication among ad hoc mobile nodes across the Internet. In the second phase packets are routed between the two connected ad hoc groups, where packets are routed inefficiently in strict hierarchical routing by using AMR gateways. Figure 1 gives an example for path selection in this network scenario. Two mobile nodes, 2.5.5 and 1.7.5, belong to two different ad hoc clusters located in a connected ad hoc network. Nodes 1.7.5 and 2.5.5 select AMR1 and AMR2 as a default gateway and use their network prefix respectively. If we use the hierarchical path which forwards the packets from AMRs instead of using the direct routing from these two different ad hoc clusters, a path selection function will be involved to choose a better routing between the hierarchical and flat routing.

### 3.4 Mobility Management

Each node in the cluster ad hoc network is affiliated with a physical subnet (SSD group) and use LSD group to form a virtual subnet. In order to connect the IP-based Internet communication, the mobile nodes need to discover the existence of the AMRs, which it belongs to, and join one of the AMRs first. The mobile nodes also must configure an IP address with the prefix of a reachable AMR (GSD group). With this partition-dependent address each mobile node can communicate with either the nodes in the same cluster network or the nodes through the Internet described as the above. The AMRs discovery could be done either by listening to an AMR Advertisement sent by AMRs (passive method) or sending an AMR REQUEST message sent by a mobile node (active method) at the initial state. In practice, both discovery methods can be combined and run in parallel. The AMR periodically sends

out advertisement, and all nodes in its radio range store this information. An AMR REQUEST, which was broadcasted by a mobile node that is not in the radio range of an AMR, can now be answered by any intermediate node with stored AMR information, thus reducing the signalling traffic. After successful AMR discovery, the mobile node registers with one of the discovered AMRs.

A mobile node which changed its subnet affiliation will notify all the nodes in its new physical and virtual subnets of its newly acquired address. We can classify two types of subnet affiliations of mobility management.

(1) Mobility within the same cluster ad hoc networks:

After moving to a new location within the same cluster a mobile node (MNa) notifies its logical neighbors and the originated AMR by sending a location update (`loc_update`). Another mobile node (MNb) which desires to communicate with the MNa will inquire at its logical neighbors about MNa by sending a location inquiry (`loc_inquire`). At least one of MNb's logical neighbors is also MNa's logical neighbor, thus, one of their mutual logical neighbors will provide MNb with MNa's address by sending `loc_track`. After tracking down MNa's address, MNb sends its data to MNa via one of their mutual logical neighbors. If MNb is out of the cluster, the routing path will be updated while the MNa's `loc_update` received by MNa's originated AMR.

(2) Mobility within the different cluster ad hoc networks:

Each MN has a permanent IPv6 home address, which has the prefix of its home network and serves as a consistent and unique identifier for the MN. While this MN moves far away from the originated register AMR and loses the connection with the members of its subnet affiliation, it may participate in the new cluster and configure a new CoA. Note that it could not only connect with its new physical and virtual subnet members but also aware of the location of this new AMR in this new cluster. Finally it sends out a BINDING UPDATE message across Internet through the new AMR to its originated register AMR (serving as Home Agent). The home agent stores the mapping between an MN's home address and CoA in a so-called binding cache and acts as a proxy for the MN. Packets addressed to a node's home address are received by the home agent and forwarded (tunneled) to the MN's CoA according to the mapping information.

Whenever a mobile node moves to another cluster and has to select a different AMR, it should become a new number of physical and virtual subnets of selected AMR, and configure a new IP address with the new prefix. Thus, when registers with a certain AMR, mobile node generates an IP care-of-address (CoA) with the IP prefix of the selected AMR. After the switch to the new AMR is completed, the mobile node generates a new IPv6 CoA with the new network prefix. The AMR can be regarded as the default gateway for virtual subnet. In other words, the entire ad hoc network is logically separated into several virtual subnets.

Unicast routing in the ad hoc side is operated in a hierarchical way based on the subnet partitioning, i.e. packets addressed to a destination in the same subnet can be directly forwarded, while packets address to a destination in a different subnet must be routed through the AMRs (even if there is a direct wireless multi-hop link between the correspondent nodes).



### 4 Simulation Analysis

To demonstrate the benefits of our adapted techniques, numerous simulations of our proposed protocol have been performed using the NS2 simulation which we enhanced the ability for mobile ad hoc nodes having communication with Internet. We also added C++ Partition Toolkit (CCPT) to determine the physical and virtual subnets to partition the nodes in a cluster ad hoc network of our proposed hierarchical network. The mobility model used in each of the simulations is in a random direction. In each ad hoc network group, nodes are initially placed randomly within a predefined 500m x 500m grid area. Each node then chooses a random direction between 0 and 360 degrees and a speed from 0 to 20 meter/second.

First, we compare the efficiency of the proposed method to Flooding, AODV, and DCA with our C++ Partition Toolkit. There are  $n$  ( $n = x * y * z$ ) nodes in our simulation grid area with  $x$  ad hoc network groups,  $y$  physical subnets and  $z$  virtual subnets and nodes can communicate each other within transmission range. Each physical subnet and virtual subnet is formed with randomly members. In the simulation study, we vary  $x$ ,  $y$  and  $z$  to compare how the network density and physical/virtual subnet sizes affect the efficiency.

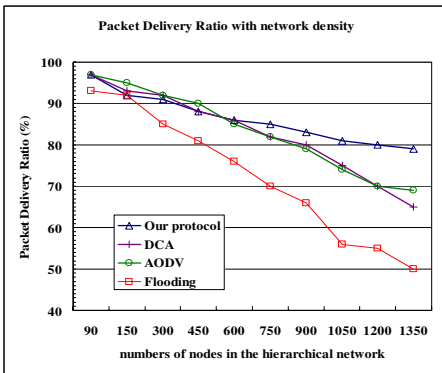


Fig. 3. The percentage of successful deliver packets

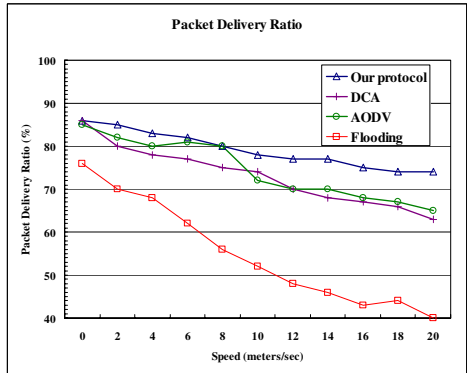


Fig. 4. Packet delivery ratio with different speed

Figure 3 shows the result for network sizes from 90 nodes to 1350 nodes with 3 to 45 virtual subnets in a fixed 3 ad hoc network groups and 10 physical subnets. As a consequence of the greater node density, our proposed method is more efficient to deliver packets to all nodes in this hierarchical network. There are some nodes moving out of transmission range, which cannot re-route successfully. It shows that the packet delivery percentage up to 78% in a dense network with 45 subnets by our proposed method. In the same condition, AODV and original flooding achieves only 69% and 50%. The main impact of the improvement is that our proposed method re-routes successfully by virtual subnets before nodes moving out of transmission range, but original flooding has too many redundant transmissions to reach destination just in time.

Figure 4 refers to a 600-node ad hoc network with 3 groups, 10 physical subnets and 20 virtual subnets. It shows the percentage of successful multicast deliver ratio for mobile nodes whose velocity varies from 0m/s to 20m/s (around 70km/h) using four different routing protocols, and all the nodes of the addressed multicast group received the packet more than 85% when the velocity is less than 8m/s for our proposed method. However, when the velocity of nodes is more than 2m/s, the packet delivery ratio of flooding will drop immensely as the velocity of nodes speed up.

## 5 Conclusions and Future Research

In this paper, we have described the adapted partition network model for large-scale ad hoc networks. The main objective of our protocol is to perform efficiently the behaviours of virtual subnets in ad hoc networks. We propose an interoperability network model integrating a self-organizing ad hoc network and the Internet/a conventional network with the same virtual subnet. Moreover, we describe a protocol to establish the virtual broadcast domains by using the IPv6 multicast-membership in ad hoc networks and perform IP-based network communications in a multi-switch backbone. Since the partition technology functions by logically segmenting the network into different broadcast domains, packets can only be delivered between fixed/mobile nodes with the same physical/virtual identity (group member). Therefore we can prevent the broadcast storm problem in MANET.

We plan to identify the suitable cache table refreshing mechanism on the proposed method in the future works. We will also generalize the clustering method to progress the behaviours of virtual subnets so that they can be applied in mobile ad hoc networks.

## References

1. Gupta and D. Ferrari, (1995), Resource Partitioning for Real-time Communication, *IEEE/ACM Trans. on Networking*, 3.5, 501–518.
2. N. Kavak, (1995), Data Communications in ATM Networks, *IEEE Network*, vol.9, no.3, May/June 1995.
3. Rajaravivarma, V.; (1997), Virtual local area network technology and applications, *Proceeding of the Twenty-Ninth Southeastern Symposium on*, 9-11 March 1997 Pages:49 – 52.
4. Internet Engineering Task Force (IETF) Mobile Ad Hoc Networks (MANET) Working Group Charter. <http://www.ietf.org/html.charters/manet-charter.html>.
5. J. Jubin and J.D. Tornow, “The DARPA Packet Radio Network Protocols,” *Proceeding of the IEEE*, Vol.75, no. 1, Jan. 1987, pp. 21-32.
6. E. Bommaiah, M. Liu, A. McAuley, and R. Talpade, “AMRoute: Adhoc Multicast Routing Protocol,” *Internet-Draft*, draft-talpade-manet-amroute-00.txt, Aug. 1998, Work in progress.
7. C. W. Wu, Y.C. Tay, and C.-K. Toh, “Ad hoc Multicast Routing protocol utilizing Increasing id-numberS (AMRIS) Functional Specification,” *Internet-Draft*, draft-ietf-manet-amris-spec-00.txt, Nov. 1998, Work in progress.

8. J.J. Garcia-Luna-Aceves and E.L. Madruga, "Core-Assisted Mesh Protocol," *IEEE Journal On Selected Areas in Communications*, Vol. 17, no. 8, Aug. 1999, pp. 784-792.
9. D. J. Baker and A. Ephremides, (1981), The architectural organization of a mobile radio network via a distributed algorithm, *IEEE Trans. Commun.*, pp. 1694-1701, Nov. 1981
10. D. J. Baker, J. Wieselthier, and A. Ephremides, (1982), A distributed algorithm for scheduling the activation of links in a self-organizing, mobile, radio network, in *Proc. IEEE ICC'82*, pp. 2F.6.1-2F.6.5.
11. I. Chlamtac and S. S. Pinter, (1987), Distributed nodes organization algorithm for channel access in a multihop dynamic radio network, *IEEE Trans. Comput.*, pp. 728-737, June 1987.
12. IEEE Std 802.1Q, 2003 Edition. (2003), IEEE standards for local and metropolitan area networks, Virtual bridged local area networks.
13. Tzu-Chiang Chiang, Ching-Hung Yeh and Yueh-Min Huang, (2004) A Forwarding Cache VLAN Protocol (FCVP) in wireless Networks, 4th IASTED International Multi-Conference on Wireless and Optical Communications - WOC 2004, July 8-10, 2004, Banff, Canada
14. Wu, C., Tay, Y., and Toh, C., (1998), Ad Hoc Multicast Routing Protocol Utilizing Increasing id-numbers (AMRIS) Functional Specification, Internet-Draft, Nov. 1998.
15. D.B. Johnson, C. E. Perkins, and J. Arkko, "Mobility support in IPv6." IETF internet draft, June 2002.
16. Jacob Sharony, "An architecture for mobile radio network with dynamically changing topology using virtual subnets", *Mobile networks and applications*, pp. 75- 86, 1996.
17. J. Xi and C. Bettstetter, "Wireless Multihop Internet Access: Gateway, Discovery, Routing and Addressing," in *Proc. International Conference on 3<sup>rd</sup> generation Wireless and Beyond*, May 2002.
18. Jeffrey S. Chase, David E. Irwin, Laura E. Grit, Justin D. Moore, and Sara E. Sprenkle, "Dynamic Virtual Clusters in a Grid Site Manager", the 12th IEEE International Symposium on High Performance Distributed Computing, 2003.
19. Zhong Fana., Siva Subramani, "An address autoconfiguration protocol for IPv6 hosts in a mobile Ad Hoc network", *Computer Communications*, Sep. 2004.
20. C. E. Perkins et al. "Ad hoc On-Demand Distance Vector (AODV) Routing," RFC 3561, July 2003, <http://www.ietf.org/rfc/rfc3561.txt>
21. D. B. Johnson and D. A. Maltz, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)," IETF Internet draft, 19 July 2004, <http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-10>
22. C.R. Lin, and M. Gerla, "Adaptive Clustering for Mobile Wireless Networks," *IEEE Journal on Selected Areas in Communications*, Vol. 15, No. 7, Sep. 1997, pp.1265-1275.
23. M. Gerla and J. T.C. Tsai, "Multicluster, Mobile, Multimedia Radio Networks," *Wireless Networks vol.I*, 1995, pp. 255-265.
24. C. C. Chiang, H. K. Wu, W. Liu and M. Gerla, "Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel," *Proc. IEEE Singapore international Conference on Networks (SICON97)*, 1997, pp. 197-211.
25. S. Basagni, "Distributed Clustering Algorithm for Ad Hoc Networks," *Proc. Int'l Symp. Parallel Architectures, Algorithms, and Networks (I-SPAN)*, June 1999, pp. 310-315.
26. M. Chatterjee, S.K. Das and D. Turgut, "WCA: A Weighted Clustering Algorithm for Mobile Ad hoc Networks", *Journal of Clustering Computing*, (Special Issue on Mobile Ad hoc Networks), Vol. 5, No. 2, April 2002, pp. 193-204.

# Exploring Small-World-Like Topologies Via SplitProber: Turning Power Laws into an Advantage in Unstructured Overlays

Xinli Huang, Wenju Zhang, Fanyuan Ma, and Yin Li

Department of Computer Science and Engineering,  
Shanghai Jiao Tong University, Shanghai, 200030, P.R. China  
{huang-xl, zwj03, fyama, li-yin}@sjtu.edu.cn

**Abstract.** Recent unstructured Peer-to-Peer systems, represented by Gnutella and Freenet, offer an administration-free and fault-tolerant application-level overlay network. While elegant from a theoretical perspective, these systems have some serious disadvantages. First, due to knowing very little about the nature of the network topology, the search algorithms operating on these networks result in fatal scaling problems. Second, these systems rely on application-level routing, which may be inefficient with respect to network delays and bandwidth consumption. In this paper, we propose a novel search algorithm, called *Split-Prober*, to explore the small-world-like topologies of these networks efficiently and scalably, by turning the power-law degree distributions in these networks to an advantage, and by making discriminative use of nodes according to their different roles in the network. As a result, we are able to reconcile the conflict of remedying the mismatch between the overlay topology and its projection on the underlying physical network, while at the same time navigating these networks with a guaranteed high efficiency and using only local knowledge as cues. Our simulation results indicate that the proposed algorithm outperforms several other well-known methods with significant performance gains.

## 1 Introduction

Providing large-scale and efficient content storage and delivery are becoming ever so important because the demand for Internet-based applications is growing at an incredible rate. Recent application-level unstructured overlay networks, such as Gnutella [1] and Freenet [2], due to their abilities to support uncoupled data allocation, complex semantic queries, self-organization in highly dynamic scenario, are considered more competent for these Internet-based applications, compared with structured overlays where both the data placement and the overlay topology are tightly controlled, such as CAN [3], Chord [4], and Pastry [5]. Nodes in these networks interact symmetrically and collectively contribute towards an administration-free and fault-tolerance decentralized storage space, by hiding the underlying dynamism and heterogeneity in these systems. While elegant from a theoretical perspective, these systems suffer from two limitations. First, due to knowing very little about the nature of network topology, the search algorithms operating on these networks, as

in the case of Gnutella, conduct a “blind” and explosive broadcast search, which results in fatal scaling problems. Second, these systems rely on application-level routing (search) that largely ignores the characteristics of the underlying physical networks, which leads to an unnecessarily large network delays with very high bandwidth consumption. We argue that for a system to function efficiently, it is important for the search algorithm to take into account both the intrinsic topological nature stemmed in this system and its projection on the underlying network.

To illustrate the topological nature behind these unstructured overlays, recent studies in measurements of Gnutella [6] and simulated Freenet networks [7], show that they contain a few nodes that have a very high degree and many with low degree following power-law distributions, that is, the probability that a node has  $k$  links is approximately proportional to  $1/k^\tau$ , where  $\tau$  is the scaling exponent. This power law in the link distribution reflects the presence of central individuals who interact with many others and play a key role in relaying information, a fact that can be exploited when designing efficient search algorithms. In addition, further study of [6] shows that Gnutella topologies demonstrate strong “small-world” phenomena: characteristic path length is comparable to that of a random graph, while the clustering coefficient stays at a very high level similar to that of a regular graph. The small-world phenomenon is pervasive in networks arising from society, nature and technology. In many such networks, empirical observations suggest that any two individuals in the network are likely to be connected through a short sequence of intermediate acquaintances [8, 9]. One network construction that gives rise to small-world behavior is where each node in the network knows its physical neighbors, as well as a small number of randomly chosen distant nodes. The latter represent shortcuts in the network. It has been shown that this construction leads to graphs with small diameter, leading to a small routing distance between any two individuals [9, 10].

The power-law degree distributions and small-world phenomena in Gnutella-like unstructured overlays can significantly impact the performance of algorithms such as those for routing or searching [9]. Therefore the existence of these properties in these networks presents an important issue to consider when designing new, more scalable application-level algorithms and protocols. As is pointed out in [11], the presence of high degree nodes in a power-law graph, so useful to speed up the search process, ironically, also worsens the search cost scaling with problems of traffic explosion and hotspots, which results in poor utilization of network bandwidth and hinders scaling. In this sense, the question we try to answer in this paper is, whether the search algorithms operating on such overlays can be coupled with considerations of these intrinsic topological nature and conduct the search process in a fashion of relatively high and but also balanced search efficiency and network utilization, using only local knowledge.

In this paper, we propose a novel search algorithm, called *SplitProber*, to explore these networks efficiently and scalably, by turning the power-law degree distributions in these networks to an advantage, and by making discriminative use of nodes according to their different roles in the network. As a result, we are able to reconcile the conflict of compensating the mismatch between the overlay topology and its projection on the underlying physical network, while at the same time navigating these networks with a guaranteed high efficiency and using only local knowledge as cues.

The main contributions of this paper are: (a) we develop a provably effective model for building unstructured overlays with desirable topological properties; (b) Based on this model, we investigate several useful techniques to realize the efficient local search in these overlay topologies and propose SplitProber, a novel local search algorithm by better exploiting the power-law degree distributions and the small-world-like structure; (c) To further improve the performance of SplitProber, we also devise several unique optional optimization mechanisms as enhancements.

The remainder of the paper is organized as follows. Section 2 provides related work. Section 3 discusses the design of the SplitProber algorithm in detail. We provide the evaluation methodology in Section 4 and then present our analytical simulation results in Section 5. In the last section, we conclude this paper and highlight some directions as future work.

## 2 Related Work

All the above facts indicates that a realistic link-distribution plays a crucial role in the effectiveness of the search strategy and that it might be a crucial ingredient in explaining the low diameter seen in messaging passing on the small-world-like unstructured overlays. To navigate these overlays efficiently, several alternative approaches have been proposed in the near past to examine the role of link distribution in overlay topology, and to utilize it for the performance optimization.

Motivated by real experiments with social networks, Kleinberg was concerned with how, given the fact that short paths existed, one could find them without complete global information. The treatment given in [9, 12] had an elegant result, but the underlying graph model did not reflect all of the important features real world problems. An important shortcoming is its particular assumption of an inverse square correlation which implies that a majority of ones contacts lie in geographical proximity. What happens if a large fraction of people know as many people outside of their city or state as inside? Would it become impossible to pass messages efficiently?

Adamic et al in [11] prefer switching from broadcasting queries to passing them only to high-degree nodes, a situation similar to that of supernodes in the FastTrack network [13]. They assumed that higher-degree peers are also capable of higher query throughputs. However without some balancing design rule, such peers would be swamped with the entire P2P signaling traffic.

The fact that the number of hops between nodes is shorter in a power-law graph implies that the broadcasting method of locating nodes and resources will return results more quickly, which inspires Yang and Garcia [14] to propose a method called iterative deepening. The method is an improvement over the default protocol when the queries can be satisfied by nodes closer than the maximum radius defined by the TTL of the default. In that case, bandwidth and processing cost are saved. Similarly to some extent, Lv et al in [15] argue that by making better use of the more powerful peers, Gnutella's scalability issues could be alleviated. Instead of its flooding mechanism, they used random walks. Their preliminary design to bias random walks towards high capacity nodes did not go as far as the ultra-peer proposals in that the indexes did not move to the high capacity nodes.

The above approaches have examined strategies for finding a node on a network knowing nothing other than the identities of one’s first and second neighbors. However, a node can learn about the network over time and adapt its search strategies. Based on this intuition, a class of algorithms called adaptive search, represented by [16] and [17], suggests that adapting the search algorithm to incorporate information learned about the network can deliver results comparable to BFS (broadcast) search while using considerably less processing power and bandwidth. However, such topological changes might destroy the merits of the power-law degree distribution and its resultant small-world characteristics.

In summary, these Gnutella-related investigations are characterized by a bias for high degree peers and very short directed query paths, a disdain for flooding, and concern about excessive load on the “better” peers. Generally, the analysis and utilization of both the overlay nature and its mapping on the physical level remains open.

### 3 SplitProber: Model and Algorithm

In this section, we detail the model we use to build desirable overlays and the main ideas of the SplitProber algorithm, including several optimization mechanisms.

As is shown analytically in [11], with high-degree seeking local search strategies in power-law graphs, the length of the average path found grows slowly as the size of the network increases, whereas the average cost in the amount of time necessary to find the path scales nearly linearly. The extremely high cost of this procedure suggests that additional clues as to the location of the target or knowledge from the network over time would be necessary to make such an approach worthwhile.

We seek the advantages of power-law degree distributions and design overlays with a small-world structure. In our model of overlay topologies, a node’s links to its neighbors are divided into two categories: *local* links and *global* links. The local links connect close nodes and the global links connect nodes chosen randomly. The fraction of links that are local, called the proximity factor ( $\delta$ ), is a key design parameter that controls the properties of the resultant overlay topology. Different values of  $\delta$  let us span the spectrum of this class of overlay topologies.

To examine the role of local links on the impact of the average search time, we give a rough theoretical complexity analysis below. If the expected degree of a neighbor is  $D$  and the fraction of random links is  $(1-\delta)$  according to the above definition of the proximity factor, then it would have approximately  $D^2$  second neighbors out of which  $(1-\delta)$  would be long range or random links so the average time it would take to reach the target if it were to never use local links is:

$$T_{\text{global only}} = \frac{N}{D^2(1-\delta)} + \beta \quad (1)$$

where  $\beta$  is the average minimum number of steps required to reach the target.

The maximum number of steps it takes if the walker starts using local links after reaching within a distance  $l$  of the target will be following:

$$T_{\text{global+local}} = \frac{N}{D^2l(1-\delta)} + \frac{\alpha l}{\delta} + \beta \quad (2)$$

where  $\alpha$  and  $\beta$  are constants. For a power law graph of power approximately 2 as is observed in Gnutella networks [18], we have the expression for  $D$  in terms of  $N$ :

$$D = \frac{N^{1/2}}{\text{Log}(N)} \quad (3)$$

Now, substituting (3) into (2) and minimizing the resulting expression with respect to  $l$  we get an upper bound for the average number of steps:

$$T_{\text{global+local}} = \sqrt{\frac{\alpha}{\delta(1-\delta)}} \text{Log}(N) + \beta \quad (4)$$

According to the results of (1) and (4), it is obvious that the most efficient search is when there are both local and random long distance links. On getting close to the target say a distance  $l$ , it becomes advantageous to use local as opposed to globally random links. This kind of topology is expected to allow one to search the power-law graph more rapidly using only local knowledge. Then here comes an important issue: how to navigate the topology in the mentioned fashion efficiently? In our proposed SplitProber algorithm below, we introduce novel techniques which operate independently and in a decentralized manner to achieve this goal.

To turn power-law degree distribution into an advantage while at the same time minimizing the search cost scaling mentioned in Section 2, we advocate directing queries towards deliberately selected high-degree neighbors with the probability of a node being chosen proportional to its degree, following  $P(N_i) \sim (\text{Degree}(N_i))^\kappa$ , where  $\kappa$  is defined as the selection strength imposed upon the high-degree nodes. Given the dynamic conditions of both the peer content and its location in a peer-to-peer network, we devise a unique adaptive better-neighbor selection scheme, by effectively coupling the “best results” scheme with the “best physical proximity” scheme. A SplitProber node first caches the nodes which had previously delivered a specified number of results in the least amount of time into its candidate neighbors list, and then updates its neighborhood by periodically evaluating the “distance” to the nodes in this list and its neighborhood according to a novel technique called Landmark Clustering [19] explained below.

**Landmark Clustering.** We pick  $m$  landmark nodes that are randomly scattered in the overlay. These landmark nodes can be part of the overlay itself or standalone. Each node measures its latencies to the  $m$  landmarks. For node  $P$ , suppose that the measured distances are  $\langle l_1, l_2, \dots, l_n \rangle$ . We then position node  $P$  in an  $m$ -dimension Cartesian space using  $\langle l_1, l_2, \dots, l_n \rangle$  as its coordinates. We call this Cartesian space, the landmark space. The intuition behind doing this is that nodes that are close to each other have similar landmark measurements, and are close to each other in the landmark space. We use the Landmark Clustering technique to re-rank all the nodes in the above both lists maintained by the peer, and then update its neighborhood by selecting the first  $k_l$  best nodes as neighbors, here  $k_l$  is the original number of the peer’s local neighbors. The aim we do this is to guarantee that all the local neighbors of a peer are really “local” both in physical proximity and in semantic proximity.



Take the dominant resource-locating application as an example, the SplitProber algorithm makes discriminative use of nodes and probes the overlay topology by splitting the search process into two distinct sub-processes elaborated as follows.

1. *Local Probing*: Due to the above two kinds of proximities, a majority of queries issued by a peer are expected to be answered successfully by or through its local neighbors. Thus the peer can forward the incoming queries to all its local neighbors using scoped-flooding with a much smaller TTL. If applied with the NoN-Lookahead local indexing mechanism (addressed later), a TTL value of 2 will result in 4 hops away from the peer, which will cover a much large fraction of the overlay due to the power-law degree distributions and the small-world-like structure.
2. *Global Probing*: To limit the explosive duplicate messages and heavy traffic load, we prefer random walks rather than flooding when probing long-range global neighbors. But much differently, we use a strategy, called Intentional Walks [20], to relay queries along global links towards the specified destination, say  $D$ . On getting close to  $D$ , the walks end and we then resort to Local Probing to complete the remaining search processes for the reasons addressed above.

Optionally, we also propose an optimization mechanism below to improve the performance of the SplitProber algorithm.

**NoN-Lookahead local indexing.** With the above procedure, when we choose a neighbor closest to the destination we do not know if it has a neighbor to take us closer to the destination. As a remedy, the clients (or just some high-degree ones) can be modified to locally keep index of the files stored by their friends, in a 2-level lookahead, their Neighbors-of-Neighbors.

## 4 Evaluation Methodology

In this section, we use simulations to evaluate SplitProber and compare its performance to three other well-known search algorithms used in unstructured overlays. Thus our simulations refer to the following four models:

1. *FLOOD*: Search using TTL-based flooding over the standard Gnutella topologies. This represents the classical Gnutella protocol.
2. *RW*: Search using random walks. This represents the recommended search techniques suggested by Lv et al [15].
3. *DS*: Search using high-degree seeking strategies, suggested by Adamic et al [11]. In this model, queries are flooded only between high-degree nodes.
4. *SplitProber*: Search using our proposed SplitProber algorithm over the desirable overlays with power-law degree distributions and small-world-like structure, according to the model we develop in Section 3.

We consider a system model where peers are organized in an overlay network. Each peer has a set of neighbors with which it communicates by message passing. Links are directed: a peer  $P$  may have another peer  $P'$  as neighbor without  $P'$  considering  $P$  as its neighbor. Traffic can however flow in both directions on the links. We use a Gnutella overlay with a  $\tau=2.1$  power-law out-degree distribution and a simple

cutoff at  $m \sim N^{1/r}$  to validate our analytical results. In the simulations, 100~500 unique files with varying popularity are introduced into the system. Each file has multiple copies stored at different locations chosen at random. The number of copies of a file is proportional to their popularity. The count of file copies is assumed to follow Zipf distribution with 2,000~4,000 copies for the most popular file and 40~80 copies for the least popular file. The queries that search for these files are also initiated at random hosts on the overlay topology. Again the number of queries for a file is assumed to be proportional to its popularity.

To make a comprehensive comparison with the other three algorithms and models, we focus on the following three aspects of performance metrics. These metrics, though simple, reflect the fundamental properties of search algorithms.

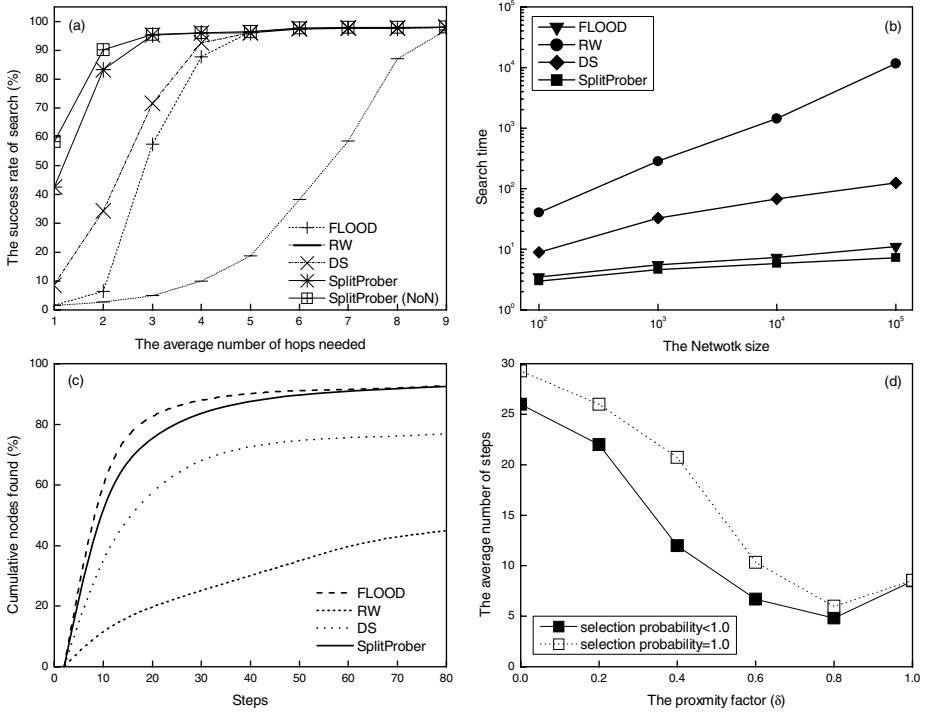
1. *Search performance*: a local gain perceived directly by a user of a system, measured along multiple dimensions such as search time, success rate of search, the cumulative nodes found, and the number of hops needed.
2. *Load aspects*: overhead of an algorithm, measured by the metrics like the average number of messages per node and the percentage of duplicate messages.
3. *Utilization of the underlying network*: a global property that is of interest to network service providers, measured by the metrics like the physical distance (latency) to the search results, and the mean stress, one of the most common definitions of the traffic load in overlay network [21].

Each experiment is repeated multiple times with different seeds to remove any bias in random number generation that is used in multiple stages of simulation.

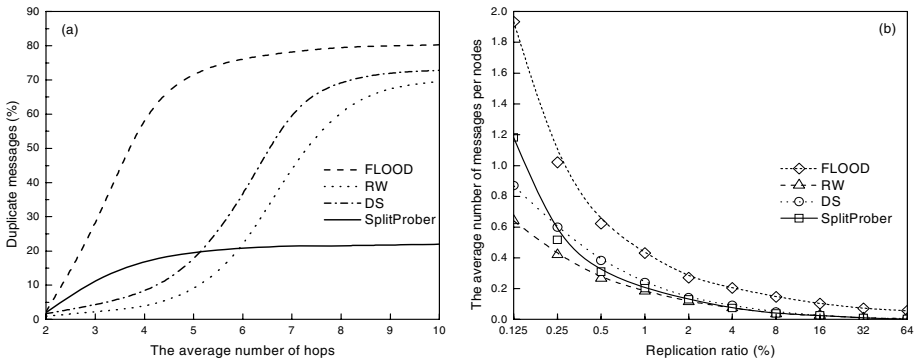
## 5 Simulation Results

In this section, we present and analyze the results of the experimental evaluation of *SplitProber*. We first start by studying the overall impact of *SplitProber* on the search performance. We then analyze the traffic load to examine the overhead of *SplitProber* compared with the other three models. Finally, we evaluate the performance gains of *SplitProber* in the utilization of underlying network.

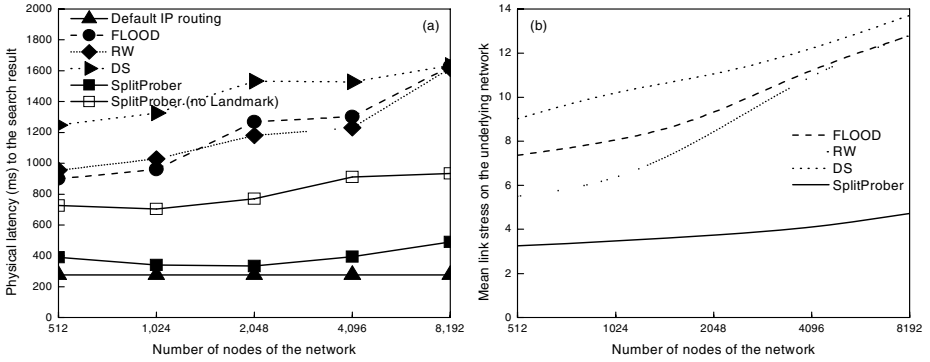
Fig.1(a) shows the search success rate as a function of the number of hops. The results clearly indicate that, with our proposed *SplitProber* algorithm, we achieve higher success rate of search even within the initial two hops, especially when the optional NoN-Lookahead local indexing mechanism is applied. The results of Fig.1(b) show that *SplitProber* can explore the overlay network with a search time almost equal to that of the flooding strategy, the recognized fastest strategy with regard to search in unstructured overlays. This performance gain can be explained by Fig.1(c), where *SplitProber* can find much more unique nodes (only a little fewer than that of FLOOD) within the same hops than those of DS and RW. To examine the role of the proximity factor  $\delta$  we introduced in the *SplitProber* design, we plot the average number of steps for a successful query as a function of  $\delta$  in Fig.1(d), the results in which show that the presence of the proximity factor with a proper value (for example, here  $\delta=0.8$ ) will contribute greatly to speeding up the search process, and that it is also more beneficial to exploit the high-degree neighbors with a probability than in a simple determinate way used by DS.



**Fig. 1.** The comparisons of search performance. (a) The success rate of search as a function of the number of hops. (b) Scaling of the average node-to-node search cost. Here the power-law exponent  $\tau=2.1$ . (c) Cumulative distribution of nodes seen vs the number of steps. (d) The average number of steps for a successful query as a function of the proximity factor, when selecting high-degree neighbors determinately or with a probability addressed in Section 3.



**Fig. 2.** The comparisons of the traffic load generated by different algorithms. (a) The percentage of duplicate messages as a function of the average number of hops traveled. (b) The average load on per node as a function of the file replication ratio.



**Fig. 3.** The comparisons of different algorithms in the performance aspects of the utilization of underlying network. (a) The average physical distance (latency) to the search result as a function of the network size. (b) The mapping of traffic load of the overlay topology on the underlying physical links, measured by the mean link stress on the underlying network as a function of the network size.

Fig.2 plots the simulation results of traffic load. We can see from Fig.2(a) and Fig.2(b) that, SplitProber produces not only a much smaller fraction of duplicate messages in the network but also fewer messages on per node, which means that the algorithm can generate lower traffic load and then distribute it more evenly across the network, with as few duplicate messages as possible.

As for the aspect of the network utilization, we can see from Fig.3 that our algorithm can make better use of the knowledge of underlying network, by dynamically optimizing the neighborhood quality to reduce the distance to search result (see Fig.3(a)), and by mapping more logical links to local physical links (see Fig.3(b)). These results further verify the significant performance gains of our proposed SplitProber algorithm.

## 6 Conclusions and Future Work

The unstructured overlay networks, while elegant from a theoretical perspective, have some serious disadvantages. First, due to knowing very little about the nature of the network topology, the search algorithms operating on these networks result in fatal scaling problems. Second, these systems rely on application-level routing, which may be inefficient with respect to network delays and bandwidth consumption. To challenge these situations, a novel local search algorithm special for these overlays, called SplitProber, is proposed in this paper. The main idea of SplitProber is to turn the power-law degree distributions in these networks into an advantage and make discriminative use of neighbor nodes according to their different roles in the network. We first develop a provably reasonable model to construct desirable overlays with power-law degree distributions and small-world-like structure. Based on this model, SplitProber probes these networks by splitting the search process into two distinct sub-processes: Local Probing and Global Probing, using two kinds of modified search strategies respectively and several optional optimization mechanisms. The simulation results from extensive experiments justify significant performance gains of SplitProber, compared with the other three well-known solutions.

Our work in this paper mainly focuses on the search process in unstructured overlays, without considering the download process. A fact that increasing files in networks are large-sized (eg., multimedia files) [22] underscores the significance of decentralized multimedia sharing applications. Accordingly, our further work will study how to incorporate effective techniques related to large-sized (and even real-time) file download process into our algorithm and make it be more practical.

## References

1. Gnutella. <http://www.gnutella.wego.com>
2. Clarke, I., et al. "Freenet: A Distributed Anonymous Information Storage and Retrieval System", in International Workshop on Design Issues in Anonymity and Unobservability, New York, USA, 2001
3. Ratnasamy, S., et al. "A Scalable Content-Addressable Network", in ACM SIGCOMM, San Diego, CA, USA, 2001
4. Stoica, I., et al. "Chord: A scalable peer-to-peer lookup service for Internet applications" in ACM SIGCOMM, San Diego, CA, USA, 2001
5. A. Rowstron and P. Druschel. "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems", International Conference on Distributed Systems Platforms (Middleware), Nov 2001.
6. Mihajlo A. Jovanovic, Fred S. Annexstein, Kenneth A. Berman. "Modeling Peer-to-Peer Network Topologies through Small-World Models and Power Laws", in Proc. of IX Telecommunications Forum Telfor, Belgrade, November 2001
7. T. Hong. "Peer-to-Peer: Harnessing the Benefits of a Disruptive Technology", edited by A. Oram (O'Reilly, ADDRESS, 2001), Chap. 14, pp. 203-241
8. D.J.Watts. "Small-worlds: The Dynamics of Networks between Order and Randomness", Princeton University Press, 1999
9. J. Kleinberg. "The small world phenomenon: an algorithmic perspective", Cornell Computer Science Technical Report 991776, 2000
10. D. J. Watts and S. H. Strogatz. "Collective dynamics of 'small-world' networks", Nature 393, 440-442, 1998
11. Adamic, L. A., Lukose R. M., Puniyani A. R., and Huberman B. A. "Search in power-law networks", <http://www.parc.xerox.com/istl/groups/iea/papers/plsearch>, March, 2001.
12. J. Kleinberg. "Navigation in a small world", Nature, 406, 2000
13. KaZaa Media Desktop, Sharman Networks Ltd., <http://www.kazaa.com>, 2001
14. B. Yang and H. Garcia-Molina. in Proc. of ICDCS 2002
15. Lv, Q., P. Cao, E. Cohen, K. Li, S. Shenker. "Search and replication in unstructured peer to peer networks", In Proc. of the 16th international conference on super-computing, Jun. 2002
16. V. Cholvi, P. Felber, E.W. Biersack, "Efficient Search in Unstructured Peer-to-Peer Networks", in European Transactions on Telecommunications, Special Issue on P2P Networking and P2P Services, Volume 15, Issue 6, 2004
17. Yang, B., P. Vinograd and H. Garcia-Molina. "Evaluating GUESS and Non-Forwarding Peer-to-Peer Search", in Proc. of ICDCS 2004
18. Clip2, <http://www.clip2.com/gnutella.html>, 2000
19. Zhichen Xu, Mallik Mahalingam and Magnus Karlsson. "Turning Heterogeneity into an Advantage in Overlay Routing", in Proc. of INFOCOM'03, 2003
20. Amit R Puniyani, Rajan M Lukose, Bernardo A Huberman. "Intentional Walks on Scale Free Small Worlds", arXiv: cond-mat/0107212 v1, 11 July 2001
21. M. Ripeanu, et al, "Mapping the Gnutella Network: Properties of Large Scale Peer-to-Peer Systems and Implications for System Design," IEEE J. on Internet Computing, 2002
22. Saroiu, S., et al. "An Analysis of Internet Content Delivery Systems", In Proc. of the Fifth Symposium on Operating Systems Design and Implementation, Boston, MA, Dec. 2002

# Efficient Uplink Scheduler Architecture of Subscriber Station in IEEE 802.16 System\*

Woo-Jae Kim<sup>1</sup>, Joo-Young Baek<sup>1</sup>, Sun-Don Lee<sup>1</sup>, Young-Joo Suh<sup>1</sup>,  
Yun-Sung Kim<sup>2</sup>, and Jin-A Kim<sup>2</sup>

<sup>1</sup> Department of Computer Science and Engineering,  
Pohang University of Science and Technology (POSTECH),  
San 31, Hyoja-Dong, Nam-Gu, Pohang 790-784, Korea  
{hades15, nalsunia, sdonlee, yjsuh}@postech.ac.kr

<sup>2</sup> WiBro Technology Lab., Telecommunication R&D Center,  
Samsung Electronics Co., LTD,  
Maetan-Dong, Paldal-Gu, Suwon 442-742, Korea  
{tseliot, jin0420}@samsung.com

**Abstract.** The IEEE 802.16, broadband fixed wireless access standard, defines four service classes, USG, rtPS, nrtPS and BE on high speed wireless networks. To guarantee the QoS requirement of these classes, the subscriber station and base station require scheduling architecture and algorithm. However, the IEEE 802.16 does not define any scheduling architecture or algorithm, and the most existing scheduling mechanisms only focus on working at the BS. In this paper, we propose two types of scheduling architecture working at the SS. In the one-level scheduler, we use a flow queue and class queue by differentiating flows with their class priority. The two-level scheduler can provide more organized QoS service with complementing the one-level scheduler. Adapting these architectures makes scheduler efficiently control all types of traffic defined in the IEEE 802.16. In the proposed architecture, any scheduling algorithms such as SCFQ and EDF can be applied. We evaluate the proposed scheduling architecture by simulation. The results of the simulation show that our proposed architecture can use the bandwidth efficiently.

## 1 Introduction

Broadband Wireless Access (BWA) is emerging as a last mile broadband access technology with several advantages: rapid deployment, high scalability, low maintenance and upgrade costs, and granular investment to match market growth [1]. BWA systems are designed to support quality of service (QoS) for real time applications such as video conference, video streaming, and voice over IP. The newly developed IEEE 802.16 standard is one of the BWA systems receiving wide attention from the industry and researchers.

---

\* This work was supported in part by the Samsung Electronics and protected by the patent (10-2005-0036409) which is assigned by the Samsung Electronics.

The IEEE 802.16 standard specifies QoS signaling mechanisms (per connection or per station) such as bandwidth requests and bandwidth allocation. To support QoS, the IEEE 802.16 standard uses the concept of service flow. The upstream service flow types defined in IEEE 802.16 are Unsolicited Grant Service (UGS), Real-Time Polling Service (rtPS), Non-Real-Time Polling Service (nrtPS), and Best Effort (BE). It also specifies the transmission grant mechanisms, such as Grants per Connection (GPC) and Grants per Subscriber Station (GPSS). When a SS (Subscriber Station) requests a bandwidth to the BS (Base Station) in the GPC mode, the BS allocates the requested bandwidth to the SS per connection basis. This means the SS can send only the packets belong to the connection granted by the BS. Therefore, the SS does not need a scheduler to schedule its multiple uplink connections since the uplink transmissions are scheduled by the BS. On the other hand, the SS should have a scheduler when it operates in the GPSS mode. When the BS receives a bandwidth request from the SS in the GPSS mode, it allocates the requested bandwidth to the SS regardless of the number of connections of the SS. Thus, the SS should schedule its multiple uplink connections to efficiently utilize the allocated bandwidth. However, the scheduler architecture and algorithm that determine the uplink and downlink bandwidth allocation and the packet scheduling are not defined in the IEEE 802.16 standard [2].

There are some research results on the scheduling architectures and algorithms of the IEEE 802.16 standard [3-5]. However, most of them focus on the scheduler architectures and algorithms at the BS side while those at the SS side are left unaddressed or just suggest some conceptual scheduling algorithms with no validations. In this paper, we propose uplink scheduling architectures for SSs to efficiently utilize the allocated uplink bandwidth in the GPSS mode: one-level scheduling and two-level scheduling schemes. In the one-level scheduling scheme, only one scheduler exists that manipulates multiple output queues. The two-level scheduling schemes consists of five schedulers and multiple queues - four class schedulers for each service class and one aggregate scheduler for coordinating the class schedulers and sharing the bandwidth among them. We evaluate the proposed scheduling architectures by simulation. The simulation results show that the proposed scheduler utilizes the allocated bandwidth more efficiently than other schedulers.

## 2 IEEE 802.16 QoS Mechanism

In the IEEE 802.16 standard, there are two channels between a SS and BS: uplink channel (from the SS to the BS) and downlink channel (from the BS to the SS). The downlink channel is the broadcast channel, while the uplink channel is shared by multiple SSs. The frame size is fixed and a frame is consist of uplink and downlink subframes. The BS dynamically determines the duration of uplink and downlink subframes based on its scheduling algorithm. The downlink data transmission is relatively simple because the BS is the only transmitting station during the downlink period. The data packets are broadcast to all SSs and a

SS picks up only the packets destined to it. In the uplink subframe, the BS determines the number of time slots that each SS will be allowed to transmit. This information is broadcast by the BS through the uplink map message (UL-MAP) at the beginning of each frame. After receiving the UL-MAP, each SS knows how long and when it can send data.

In the IEEE 802.16 scheduling architecture, two modules (The Admission Control (AC) module and Uplink Packet Scheduling (UPS) module) reside in the BS for supporting QoS. The AC module handles connection establishments including handshaking connection requests and responses on starting a communication between a SS and BS. On the connection establishment time, bandwidth allocation is performed between the BS and the SS. When the SS requests bandwidth based on its backlogged traffic, the UPS module in the BS sees the request on per station basis and then grants the requested bandwidth to the SS. This type of bandwidth association is called GPSS mode. The standard specifies another bandwidth association scheme called GPC mode. If GPC mode is used, the BS grants bandwidth per connection so that it guarantees the QoS. So the SS scheduler does not need to maintain QoS among its connections and control for sharing the bandwidth among the connections for fairness. Thus, we assume GPSS scheme is used in this system.

The UPS lets the flows that destined to the same destination have the same connection ID in the SS. The connection classifier in the SS classifies data packets with each connection ID and let all packets generated from application layer get into the proper queue. Then the scheduler of the SS picks up the data packet from the queues and transmits it in the appropriate time slots as indicated in the UL-MAP sent by the BS [6]. Based on this architecture, the IEEE 802.16 standard [2] defines the following four categories of service flows to fulfill each flow's various QoS requirements.

**Unsolicited Grant Service (UGS):** This service is designed for supporting constant bit-rate real-time flows such as voice over IP. To service this kind of traffic, the BS provides fixed size unsolicited data grants on a periodic basis.

**Real Time Polling Service (rtPS):** The rtPS is for real-time VBR-like flows that generate variable bit-rate data in a period such as MPEG video. The applications belong to this category receive specific bandwidth for not missing deadline.

**Non-Real Time Polling Service (nrtPS):** This service supports for non-real time flows which are variable size data and requires delay-tolerant data stream service, such as servicing high bandwidth FTP. The nrtPS offers periodic timely unicast request opportunities, so the SS should contend to request bandwidth to the BS.

**Best Effort (BE) Service:** The service BE is for best effort traffic such as HTTP. There is no QoS guarantee. The applications of this category receive the remained bandwidth after the bandwidth is allocated to the previously mentioned three service flows.



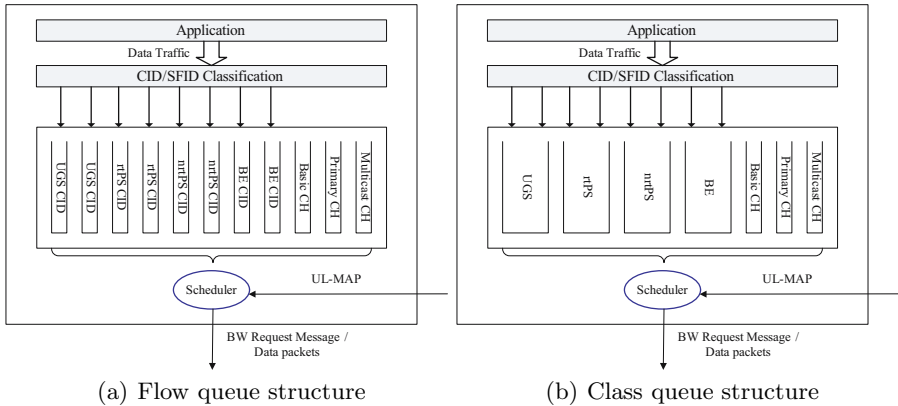


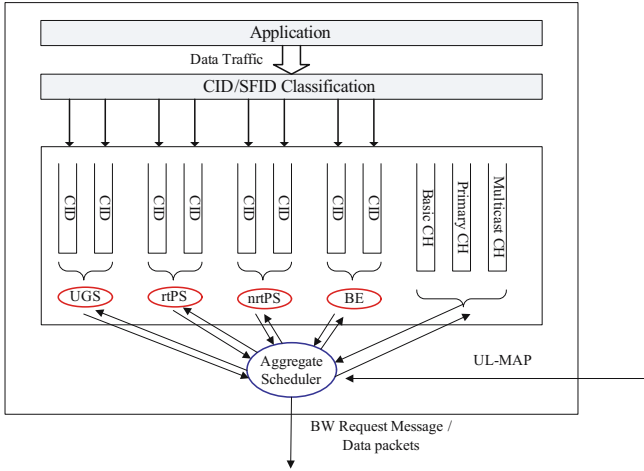
Fig. 1. The architecture of one-level scheduling scheme

### 3 Proposed Scheduler Architecture

#### 3.1 Normal Scheduler: One-Level Scheduling Scheme

To fulfill various QoS requirements of flows, a scheduling algorithm is required to schedule packets among multiple flows. The general scheduling algorithm, such as SCFQ [7], EDF [8, 9], etc., has one criteria to set a priority of each packet. Therefore one-level scheduling scheme, which is shown in Figure 1, treats all flows equally no matter which class they belong to. The scheduling algorithm used in one-level scheduler has multiple queues and, in general, one queue corresponds to one service flow. But if the scheduler makes one queue whenever new connection is established (one queue per one service flow), the queue management operation may lead to much overhead at the SS. To reduce this overhead, we can use a static queue allocation method. Because the IEEE 802.16 defines only four classes of service flows, it is enough to the scheduler with four queues (one queue per one service class). In this paper, we consider the two cases and refer them to as *flow queue* structure and *class queue* structure, respectively.

There are tradeoffs between the flow queue and class queue. If a scheduler uses a *flow queue* structure, the queue management operation may cause much overhead. But the priority of packets of each flow is not affected by other flows because all packets in the same queue belong to only one flow. Therefore a scheduler can guarantee fairness among all flows. As you can see in Figure 1(a), all flow queues are treated equally without discriminating a flow's priority when the scheduler selects a packet to transmit. If the flows have similar characteristics, it has no problem to use traditional scheduling algorithm. However, if the type of flows is different, the traditional scheduling algorithm may not schedule properly because of its lack of scheduling criteria. The *class queue* divides incoming packets into four categories based on their service classes. When a packet arrives at MAC layer, the packet is inserted into an appropriate queue among four class



**Fig. 2.** The architecture of two-level scheduling scheme

queues (see Figure 1(b)). The static allocation of queues can reduce the queue management overhead, but the packets of flows that belong to the same class are inserted into one queue. Therefore an early arrived packet is served earlier than other packets within the same class queue, and the scheduler cannot guarantee fairness among flows belong to the same class.

In these two queue architectures, the scheduler cannot satisfy QoS requirements of each flow, because the scheduling algorithm has only one criteria but flows have different characteristics and QoS requirements. For example, there are two flows: a rtPS class flow requiring guaranteeing service delay bound, and a nrtPS class flow. Suppose that the one-level scheduler uses the SCFQ scheduling algorithm and a class queue. When a packet arrives at queues, the scheduler classifies a class of the packet, calculates its weight, and inserts it into an appropriate class queue. But the SCFQ algorithm does not consider a delay and delay variation of flows, and thus it cannot guarantee a delay bound requirements of flows belong to the rtPS class. When the EDF algorithm is applied, it cannot guarantee a required bandwidth of flows belong to the nrtPS class because it has no methods to guarantee bandwidth. In the class queue structure, the EDF algorithm also cannot guarantee the delay bound of flows belong to same class because all packets on same class are inserted into one queue. If the flow queue is applied, the problems stated above except guaranteeing delay bound among flows belong same class in EDF algorithm, cannot be solved.

### 3.2 Efficient Scheduler: Two-Level Scheduling Scheme

The two-level scheduling scheme is designed for serving packets as well as satisfying specific QoS requirements of each flow. The two-level scheduling scheme can support various QoS requirements of each flow using a hierarchical scheduler ar-

chitecture. As you can see in Figure 2, two level scheduling scheme consists of aggregate scheduler and four class schedulers. The aggregate scheduler distributes bandwidth to each class scheduler when the BS allocates bandwidth to the SS. When the class scheduler receives bandwidth from the aggregate scheduler, it can serve packets of its flow queues in order of class priority. The class scheduler uses flow queue for classifying packets and various scheduling algorithms can be applied to each class scheduler. In each class scheduler, the two-level scheduling scheme chooses an efficient scheduling algorithm which can guarantee QoS requirements of each service class. Therefore, the two-level scheduling scheme has multiple scheduling criteria and schedules well packets based on appropriate QoS requirements per class. In each class scheduler, the backlogged packets have similar QoS properties and constraints, and the class scheduler only schedules flows having similar QoS constraints. Therefore the flows of each service class can receive more fair scheduling service than the one-level scheduling scheme. The aggregate scheduler distributes bandwidth to each class with proportional ratio based on class priority and the amount of backlogged packets in each class.

### 3.3 Considering the Aggregate Scheduler

Each class scheduler only transmits packets to the BS based on the allocated bandwidth from the aggregate scheduler. If a class scheduler does not receive sufficient bandwidth from the aggregate scheduler, it may not guarantee the QoS requirement. Therefore the distributing methods of the aggregate scheduler should be carefully designed. For efficient distributing of allocated bandwidth from the BS, the aggregate scheduler should know the amount of backlogged packets of each class scheduler. Because the class scheduler knows the amount of backlogged packets in its queues, the aggregate scheduler retrieves that value from the class scheduler before it distributes the bandwidth.

One possible distribution method is that the aggregate scheduler provides an opportunity for using bandwidth to the UGS class scheduler first. After serving all backlogged packets of the UGS class, the remaining bandwidth is distributed to the rtPS class scheduler. After that, the remaining bandwidth is distributed to the nrtPS and BE class scheduler. Using this distribution method, aggregate scheduler differentiates each class based on the priority of service classes. This method is simple but rtPS or nrtPS flows may not receive sufficient bandwidth. Another possible method is that the aggregate scheduler divides bandwidth into four pieces to satisfy proportional fairness of service classes, and distributes each bandwidth piece to each class scheduler. This method can prevent the starvation of a relative low priority class such as the nrtPS. We compare the performance of these two distribution methods by simulation in the following section.

## 4 Performance Evaluations

### 4.1 Simulation Environments

We use a simulation to investigate the effect of scheduling architectures (one-level *vs.* two-level) and types of queue (flow queue *vs.* class queue). We implement a

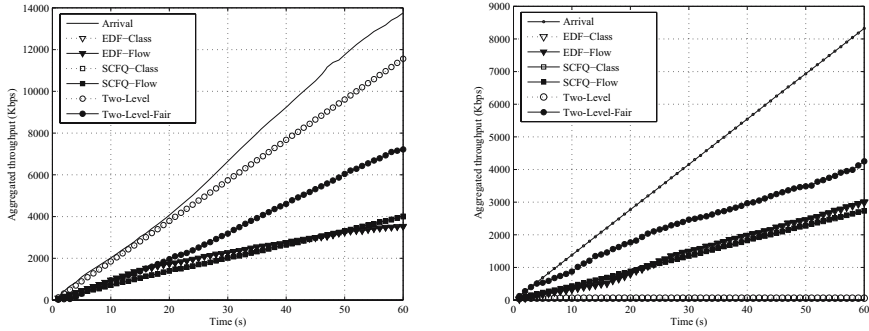
simulator using the C programming language including a BS module, a SS module, a network module, etc. In the BS module, we use a simple round-robin scheduling algorithm to allocate a requested uplink bandwidth because we focus on the efficiency of the scheduling architecture in the SS. The BS module also manages all SSs in its cell and we evaluate the performance of scheduling architectures in one of SSs. When the BS module allocates a requested bandwidth, the SS sends packets according to its scheduling algorithms. In this paper, we use the SCFQ and EDF algorithms in one-level and two-level architectures because of their simplicity and efficiency. Therefore, we also implement the SCFQ and EDF algorithms. We assumed that the size of a queue is unlimited, so there is no packet drop caused by the buffer overflows. In the channel model, we assume that the channel is static and the QPSK modulation scheme which is the default modulation method in the IEEE 802.16 uplink packet transmission is used.

We compare the performance of each scheduling architecture when there are 10 SSs in the cell. A SS has multiple uplink flows belong to each service class. In this simulation, we set the number of flows at each SS is 4 (one flow per one service class) or 8 (two flows per one service class). The simulation runs during 60 simulation seconds, and five simulation results are averaged. In one-level scheduler, we use SCFQ and EDF algorithms with flow and class queues (SCFQ-Flow, SCFQ-Class, EDF-Flow, EDF-Class). In two-level scheduler, the FIFO algorithm is used in the UGS and BE class scheduler, and the EDF and SCFQ algorithms are used in the rtPS and nrtPS class scheduler. In the aggregate scheduler, a priority based distribution method (Two-Level) and a proportional fair distribution method (Two-Level-Fair, which distributes the bandwidth to rtPS and nrtPS class scheduler with 2:1) are used. Therefore the performances of 6 scheduler architectures are compared in this simulation.

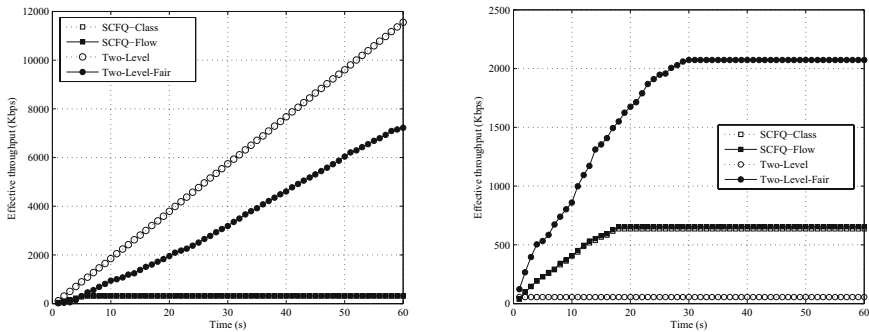
The performance metrics of our simulation results are the aggregate and effective throughput. The aggregate throughput is a cumulative throughput. The effective throughput is an amount of packets which is meaningful at the receiver. When the SCFQ algorithm is used to schedule packets belong to a rtPS class, packets which arrive at queues lately, may remain in the queue because of limited bandwidth. In this case, remaining packets are transmitted at next allocated frame from the BS. However, if these packets do not arrive at the receiver in the required delay bound, they are just dropped and thus waste wireless bandwidth. We refer this wasted bandwidth to useless throughput, and the effective throughput is derived from subtracting the useless throughput from the aggregate throughput.

## 4.2 Simulation Results

Figure 3 shows the throughputs of rtPS and nrtPS traffics when there are 4 flows in the SS. During our simulation, the compared scheduling algorithms can serve UGS traffics well, so we do not include the throughput results of UGS traffics in this paper. Also, because the traffics of the BE class have no requirements of QoS, throughput results of the BE class are not included. Figures 3(a) and 3(b) show the aggregated throughput of rtPS and nrtPS traffics in one flow per one



(a) Aggregated throughput of rtPS traffics (b) Aggregated throughput of nrtPS traffics



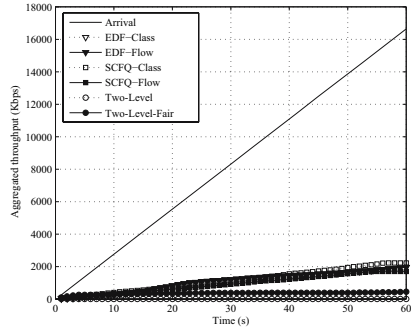
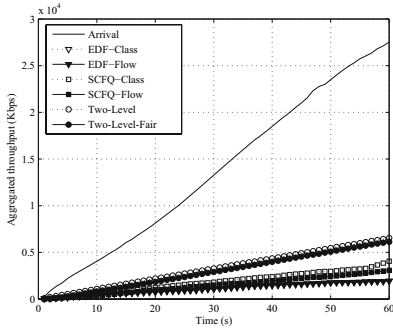
(c) Effective throughput of rtPS traffics (d) Effective throughput of nrtPS traffics

**Fig. 3.** Throughput of rtPS and nrtPS traffics (4 flows)

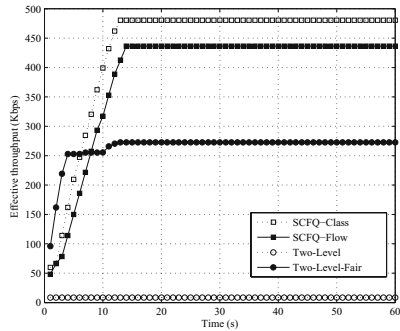
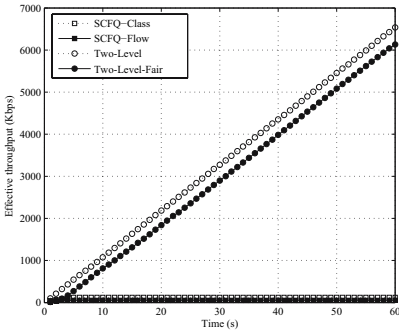
class case. When there is one flow per class, the class queue and flow queue is the same. Therefore, plots of the SCFQ-Class and SCFQ-Flow, EDF-Class and EDF-Flow are overlapped.

Comparing the throughput between two traffics, we can see that the two-level scheduler achieves the highest throughput for the rtPS traffics, but lowest throughput for the nrtPS traffics. This means that all the remaining bandwidth is used for serving rtPS traffics after serving UGS traffics. However, the two-level-fair can efficiently share the remaining bandwidth between rtPS traffics and nrtPS traffics. We also derive the useless throughput of each scheduler, but do not include in this paper. In the useless throughput plots, we can see that there are no useless throughput in two-level and two-level-fair schedulers. This means that the two-level-fair scheduler efficiently distributes the allocated bandwidth to rtPS and nrtPS traffics with no throughput degradation.

The SCFQ based algorithm (SCFQ-Class and SCFQ-Flow) uses the weighted sharing of the allocated bandwidth from the BS. In our simulation, the SCFQ uses the same weight to each service class. When the application generates more data than the scheduler can serve, data packets are stored in the queue of the



(a) Aggregated throughput of rtPS traffic (b) Aggregated throughput of nrtPS traffic



(c) Effective throughput of rtPS traffic (d) Effective throughput of nrtPS traffic

**Fig. 4.** Throughput of rtPS and nrtPS traffic (8 flows)

SS and served at the next interval. In this case, many data packets experience much delay resulting drops at the destination because the delay constraints of the rtPS traffic are violated. But the SCFQ-based algorithm just sends data packets in enqueueing order because there is no delay consideration in the SCFQ-based algorithm. So we derive the effective throughput results as shown in Figures 3(c) and 3(d). The effective throughput is derived by subtracting the useless throughput from the aggregated throughput. In these figures, we also know that the two-level-fair scheduler can efficiently use the bandwidth sharing between rtPS and nrtPS traffic although the throughput of rtPS traffic is lower than the two-level algorithm.

Figure 4 shows the aggregate and effective throughput of rtPS and nrtPS traffic when there are 8 flows. In Figures 4(a) and 4(b), the aggregate throughput of rtPS traffic in the two-level scheme is almost same as that of two-level-fair, but the aggregate throughput of nrtPS traffic in the two-level scheme is almost zero. In Figures 4(c) and 4(d), we can see that the effective throughput of rtPS traffic in the two-level-fair scheme is almost same as that of two-level, and the two-level-fair scheme also can support the nrtPS traffic much well than the two-level scheme.

## 5 Conclusions

The IEEE 802.16 defines four service classes, USG, rtPS, nrtPS and BE. To guarantee the required QoS of these classes, the SS and BS should have a scheduling architecture and algorithm which are not defined in the standard.

In this paper, we propose two types of scheduling architectures: One-level scheduler and Two-level scheduler. Between them, the two-level scheduler can provide more organized QoS service than the one-level scheduler. Adapting these architectures makes scheduler control efficiently all types of traffic defined in the IEEE 802.16. We evaluate the proposed scheduling architecture by simulation. The results of the simulation show that the proposed scheduler can use the bandwidth efficiently than other schedulers. In the two-level scheduler, the bandwidth distribution method of the aggregate scheduler is a critical factor on the performance and should be carefully designed.

## References

1. GuoSong Chu, Deng Wang, and Shunliang Mei, *A QoS Architecture for the MAC Protocol of IEEE 802.16 BWA System*, IEEE International Conference on Communications, Circuits and Systems and West Sino Expositions, vol. 1, pp. 435-439, Jul. 2002.
2. IEEE 802.16-2004, *Air Interface for Fixed Broadband Wireless Access Systems*
3. Mohammed Hawa and David W. Petr, *Quality of Service Scheduling in Cable and Broadband Wireless Access Systems*, IEEE International Workshop on Quality of Service, pp. 247-255, May 2002.
4. G.Nair, J.Chou, T.Madejski, K.Perycz, D.Putzolu and J.Sydir, *IEEE 802.16 medium access control and service provisioning*, Intel Technology Journal, vol. 8, no. 3, pp. 213-228, Aug. 2004.
5. Dong-Hoon Cho, Jung-Hoon Song, Min-Su Kim, and Ki-Jun Han, *Performance Analysis of the IEEE 802.16 Wireless Metropolitan Area Network*, IEEE International Conference on Distributed Frameworks for Multimedia Applications, 2005.
6. Kittiwongthavarawat and Aura Ganz, *Packet scheduling for QoS support in IEEE 802.16 broadband wireless access systems*, International Journal of Communication System, vol. 16, pp. 81-96, John Wiley & Sons, 2003.
7. S.Jamaloddin Golestani, *A Self-Clock Fair Queueing Scheme for Broadband Applications*, IEEE INFOCOM'94, Jun. 1994.
8. L.Georgiadis, R.Guerin, and A.Parekh, *Optimal multiplexing on a single link: delay and buffer requirements*, IEEE Transactions on Information Theory, vol. 43, no. 5, pp. 1518-1535, Sep. 1997.
9. J.Liebeherr, D.Wrege, and D.Ferrari, *Exact admission control for networks with a bounded delay service*, IEEE/ACM Transactions on Networking, vol. 4, no. 6, pp. 885-901, Dec. 1996.

# A Survey of Anonymous Peer-to-Peer File-Sharing

Tom Chothia and Konstantinos Chatzikokolakis

Laboratoire d'Informatique, École Polytechnique, 91128 Palaiseau Cedex, France  
{tomc, kostas}@lix.polytechnique.fr

**Abstract.** This paper provides a survey of searchable, peer-to-peer file-sharing systems that offer the user some form of anonymity. We start this survey by giving a brief description of the most popular methods of providing anonymous communication. These include the Ants protocol, Onion routing, Multicasting, MIXes and UDP address spoofing. We then describe a number of implemented systems based on one, or a combination of, these methods. Finally, we discuss possible attacks on the anonymity of these systems and give examples of particular attacks and defences used by the systems we describe.

## 1 Introduction

File-sharing is a hot topic in academic circles, in the open source community and in the media, but there has been very little exchange of ideas between these groups. There are a number of peer-to-peer systems that offer some kind of anonymity. However, when talking about anonymous systems it is vital to be precise about what is anonymous, from whom, under what conditions, and exactly how anonymous. In this paper we discuss both the theory of anonymity and the implemented file-sharing systems.

The majority of anonymous peer-to-peer systems are “friend-to-friend” networks. These are peer-to-peer networks in which each peer (node) only connects to a small number of other, known nodes. Only the direct neighbours of a node know its IP address. Communication with remote nodes is provided by sending messages hop-to-hop across this overlay network. Routing messages in this way allows these networks to trade efficient routing for anonymity. There is no way to find the IP address of a remote node, and direct neighbours can achieve a level of anonymity by claiming that they are just forwarding requests and files for other nodes. These systems offer anonymity against an attacker that is a single node inside the system and that is looking for the IP address of someone who is searching for, or offering, a file.

In these systems every node has a pseudo address that can be used for communication. It is easy to find the pseudo address of any node that is sending or receiving a file, but it is hard to link these pseudo addresses with the node’s real IP addresses. Furthermore, a node may stop using a pseudo address at anytime, independently make a new one or even have more than one.



There is a danger that the attacker will be able to link the pseudo address and the IP address of their direct neighbours, and thus find out which files the neighbour is requesting or offering. Some systems contain faults that leak this information while others allow an attacker to be up to 50% certain of their neighbour's pseudo address. This is more of a problem than it seems because, in the interests of growing the network, most systems make it easy for anyone to join at any point. It would be possible for an attacker to try random IP addresses until it finds someone running the protocol, and then negotiate a connection. None of the current systems try to make it hard for an attacker to work out whether or not someone is running the file-sharing software.

The level of anonymity a system offers usually degrades as more attackers join the network. Successful attacks are much easier if the attackers can choose where they join a system, particularly if they can surround a node. If the attacker knows how the peers are connected to each other, the systems offer very poor protection. There is also poor protection against attacks based on measuring the time a node takes to respond. Most systems offer no protection against an attacker that can observe all network communications, although it is possible that a MIX [Cha81] based file-sharing system could be effective.

In Section 2, we detail a number of theoretical ideas and protocols that form the basis for most types of anonymity. Section 3 catalogues implemented peer-to-peer systems that offer some kind of anonymity. We pay particular attention to systems that allow for anonymous searches and file downloading. We present a number of possible attacks against anonymous file-sharing systems in Section 4. The two most potent attacks against the current batch of implemented systems seem to be the use of multiple attackers - especially when used to surround a node - and time-based, statistical analysis attacks.

There are many interesting social, legal and economic issues related to anonymous peer-to-peer systems; too many to be covered in this survey. Many of these issues also lie outside the field of computer science. A collection of papers on these issues can be found at <http://www.inf.tu-dresden.de/~hf2/anon/>. A longer, more detailed version of this paper is available online.

## 2 Theoretical Background

When we design or analyse an anonymous system we must define what we mean by “anonymous”. Generally speaking, our purpose is to hide the relationship between an observable action (for example, a message sent across a public network) and the identity of the users involved with this action. Some questions that immediately arise are “Which identity do we want to hide?”, “From whom?” and “To what extent?”. The answers to these questions lead to different notions of anonymity.

The main agents involved in file-sharing are the *sender*, who initiates a search for a file, and the *responder* or *receiver* who answers the search query and provides the file. In peer-to-peer networks these agents are communicating through a number of *nodes* which forward the request and possibly the search data. A

*global attacker* is considered to have access to all messages that are sent over the network. These definitions lead to the following kinds of anonymity.

- Sender anonymity to any node, the responder or a global attacker.
- Responder anonymity to any node, the sender or a global attacker.
- Sender-responder unlinkability to any node or a global attacker.

It may also be useful to consider an attacker that is a combination of a global attacker, sender, receiver and any number of nodes inside the system. Pfitzmann and Hanse [PK04] provide an extended discussion on this topic. Considering the level of anonymity provided by a system, Reiter and Rubin [RR98] provide the following useful classification:

- Beyond suspicion (BS)* From the attacker’s point of view, the detected user appears no more likely to have originated the action than any other node.
- Probable innocence (ProbI)* From the attacker’s point of view, the detected user appears no more likely to have originated the action than to not to have.
- Possible innocence (PossI)* From the attacker’s point of view, there is a nontrivial probability that the detected user did not originate the action.

The following table gives a general idea of the anonymity provided by the methods discussed in the rest of this section. In many cases only a rough estimate of the anonymity guarantees is given in the corresponding papers, and the notions examined are not always the same.

Anonymity theories	Ants	Mixes	Crowds	Onion Routing	DC-nets	Multi-cast	Spoofed UDP	Freenet
S. anon. to G.A.	No	No	No	No/BS	BS	No	No	No
R. anon. to G.A.	No	No	No	No	BS	BS	No	No
S. anon. to R.	ProbI	BS	BS	BS	BS	No	ProbI	ProbI
S. anon. to N.	ProbI	No	ProbI	No/BS	BS	No	ProbI	ProbI
R. anon. to S	ProbI	No	No	No	BS	BS	No	No
R. anon. to N.	ProbI	No	No	No	BS	BS	No	No
S.-R. unlink. to N.	ProbI	BS	ProbI	BS	BS	BS	ProbI	ProbI
S.-R. unlink. to G.A.	No	BS	No	BS	BS	BS	No	No

The **Ants** protocol [GSB02] was designed for ad-hoc networks, in which nodes do not have fixed positions. In this setting, each node has a pseudo identity which can be used to send messages to a node, but does not give any information about its true identity. In order to search the network, a node broadcasts a search message with its own pseudo identity, a unique message identifier and a time-to-live counter. The search message is sent to all of the node’s neighbours, which in turn send the message to all of their neighbours until the time-to-live counter runs out. Upon receiving a message, a node records the connection on which the message was received and the pseudo address of the sender. Each node dynamically builds and maintains a routing table for all the pseudo identities it sees. This table routes messages addressed to a pseudo identity along the

connection over which the node has received the most messages from that pseudo identity. To send a message to a particular pseudo identity, a node sends a message with the pseudo identity as a “to” address. If a node has that pseudo address in its table, it forwards the message along the most used connection. Otherwise, it forwards the message to all its neighbours.

There exists no published anonymity analysis of the Ants protocol. It is widely believed that the Ants protocol provides probable innocence if a proper probabilistic time-to-live counter is used.

**Onion routing** is a general-purpose protocol [SGR97] that allows anonymous connection over public networks on condition that the sender knows the public keys of all the other nodes. Messages are randomly routed through a number of nodes called *Core Onion Routers (CORs)*. In order to establish a connection, the initiator selects a random path through the CORs and creates an onion, a recursively layered data structure containing the necessary information for the route. Each layer is encrypted with the key of the corresponding COR. When a COR receives an onion, a layer is “unwrapped” by decrypting it with the COR’s private key. This reveals the identity of the next router in the path and a new onion to forward to that router. Since inner layers are encrypted with different keys, each router obtains no information about the path, other than the identity of the following router.

There are two possible configurations for an end-user. They can either run their own COR (local-COR configuration) or use one of the existing ones (remote-COR). The first requires more resources, but the second provides better anonymity.

Onion routing has also been adapted to a number of different settings. The table gives the values for original Syverson, Goldschlag and Reeds version discuss here, the two values of sender anonymity correspond to the remote-COR (left) and local-COR (right) configurations.

**Freenet** [CSWH01] is a searchable peer-to-peer system for censorship resistant document storage. It is both an original design for anonymity and an implemented system. While it does not aim to hide the provider of a particular file it does aim to make it impossible for an attacker to find all copies of a particular file. A key feature of the Freenet system is that each node will store all the files that pass across it, deleting the least used if necessary. A hash of the title (and other key words) identifies the files. Each node maintains a list of the hashes corresponding to the files on immediately surrounding nodes. A search is carried out by first hashing the title of the file being searched for, and then forwarding the request to the neighbouring node that has the file with the most similar hash value. The node receiving the request forwards it in the same way. If a file is found, it is sent back along the path of the request. This unusual search method implements a node-to-node broadcast search one step at a time. Over time it will group files with similar title hash values, making the search more efficient.

**Return Address Spoofing** can be used to hide the identity of the sender. The headers of messages passed across the Internet include the IP address of

the sender. This address is not used by routers, so it does not have to be correct. The Transmission Control Protocol (TCP) uses this return address to send acknowledgements and control signals, but the User Datagram Protocol (UDP) does not require these controls. Simply by using the UDP protocol and entering a random return address, a sender can effectively send data and hide their identity from the receiver. Without the controls of TCP, packets are liable to loss or congestion. However, if the receiver has an anonymous back channel to communicate with the sender, it can use this to send control signals. A problem with UDP-spoofing is that such behaviour is associated with wrongdoing, and so it is often prohibited by ISPs.

A **Broadcast** can be used to provide receiver anonymity by ensuring that enough other people receive the message to obscure the intended recipient. A broadcast can be performed in an overlay network by having each node send a message to all of its neighbours, which in turn send it to all of their neighbours, and so on. If a unique identity is added to the message, nodes can delete reoccurrences of the same message and stop loops from forming. In large networks it may be necessary to include some kind of time-to-live counter to stop the message flooding the network. In anonymous systems this counter is usually probabilistic. One of the most useful methods of broadcasting is Multicasting [Dee89].

**Crowds** is an anonymous protocol for web-transactions proposed by Reiter and Rubin [RR98]. This protocol involves a group of users, called a “crowd”, each of whom wants to communicate with a corresponding web server but without revealing his identity. The idea is to randomly route each message through the crowd until one member of the crowd decides to pass it to the server. This ensures that neither the receiver nor the nodes in the system can tell who sent the message. This system requires all nodes to be connected to all other nodes, and so it scales badly to larger networks.

**MIXes** [Cha81] provide anonymity by forwarding messages from node to node, but instead of forwarding each message as it arrives, the nodes wait until they have received a number of messages and then forward them mixed up. When done correctly this can hide the sender and the receiver, as well as sender-receiver linkability from an attacker that can see the whole network. This can be done without requiring all of the nodes to consistently broadcast packets. One drawback is that each node has to hold a message until it has enough messages to properly mix them up, which might be difficult in a file-sharing system due to the asymmetrical nature of downloading files. There are many different kinds of MIX systems. The values given in the table are for the classical Mixes described here, in which end-users do not perform mixing themselves, but they communicate with a mix node in order to send a message. Protection between the mix nodes is better.

**DC-nets** [Cha88] and **XOR-trees** [DO00] use the XOR of combinations of messages. These methods provide perfect anonymity by requiring all members of the system to broadcast in every time slice.

### 3 Implemented Anonymous Systems

This section discusses implemented and publicly available systems for searchable, anonymous peer-to-peer file-sharing. The following table summarises the systems discussed in the section.

The development of real anonymous peer-to-peer systems can be more troublesome than one would at first suspect. A case in point was an anonymous peer-to-peer system known as “Winny”. The author of this system pushed it as a truly anonymous file-sharing system and file-sharers who wished to swap movies quickly picked it up. While the specification of the system was never fully released, there was soon firm evidence that the system did not really guarantee anonymity, as police arrested two of the system’s users and charged them with copyright theft. Shortly after this, the author of the software, who was a researcher in the Computer Science Department of Tokyo University, was also arrested and charged with aiding and abetting copyright theft [Ley04].

Name	Based on	Website or Paper
Ants	Ants	<a href="http://antsp2p.sourceforge.net">http://antsp2p.sourceforge.net</a>
AP3	Crowds	[MOP <sup>+</sup> 04]
APFS	Onion routing	[SLS01]
Entropy	Freenet	<a href="http://entropy.stop1984.com">http://entropy.stop1984.com</a>
Free Haven	Secret sharing and MIXes	[DFM00]
Freenet	Freenet	[CSWH01]
GNUnet	MIXes	<a href="http://gnunet.org/">http://gnunet.org/</a>
I2P	Onion routing	<a href="http://www.i2p.net/">http://www.i2p.net/</a>
Mantis	Ants and UDP spoofing	[BASM04]
Mute	Ants	<a href="http://mute-net.sourceforge.net">http://mute-net.sourceforge.net</a>
Nodezilla	Freenet	<a href="http://www.nodezilla.net">http://www.nodezilla.net</a>
Napshare	Ants	<a href="http://napshare.sourceforge.net">http://napshare.sourceforge.net</a>
Tor	Onion routing	[DMS04]
SSMP	Secret sharing and onion routing	[HLX <sup>+</sup> 05]
Waste	Friend-to-Friend	<a href="http://waste.sourceforge.net">http://waste.sourceforge.net</a>

**Mute**, **ANTS** and **Mantis** [BASM04] implement the Ants protocol. Mute uses a three-stage probabilistic time-to-live counter to avoid flooding the network. The time-to-live is reduced by a value proportional to the number of results found at a node and the number of connections the search message is forwarded to. This stops a node being flooded with too many responses, while allowing searches for less common files to go further. However, analysis of the counter may allow a statistical attack. Another point of interest in Mute is that all the probabilistic choices are fixed when a node starts running. This protects against statistical attacks by ensuring that the repetition of the same action yields no new information to the attacker.

Ants does not use a time-to-live counter. Instead, there is a chance that a packet will be dropped at any time. The Ants routing algorithm may send responding packets along more than one path, which leads to faster file downloads.

Mantis allows the searcher to exchange anonymity for download efficiency. Anonymous communication is used to search for files and to send control signals, while the data can be sent directly from the server to the client using return address spoofed UDP. To protect against attackers surrounding a node, Mantis uses a “Blender” to control access to the system.

**Anonymous Peer-to-peer File-Sharing (APFS)** [SLS01] is a searchable system with responder and sender anonymity. It is based on Onion Routing, with the addition of volunteer nodes that will act as proxies. Centralised servers are necessary to handle the searches. When a node is willing to share a file, it first picks an anonymous proxy and sends that proxy an onion route and a random identity. The node then sends the server the names of the files it is willing to offer, the name of the proxy and the random identity used to identify the connection.

**Free Haven** [DFM00] is an anonymous publishing system. It is made up of a number of servers - known as servnets - which agree to store and provide documents for anyone. The identities of these servnets are publicly known. All communications are made over an external MIX-based communication layer. When a publisher wants to publish a file on Free Haven it breaks it into a number of parts using Rabin’s information dispersal algorithm and sends each part to a different servnet. When a reader wants to download a file it must first find the hash of the file it is searching for and send this to a servnet. The servnet broadcasts the request to the other servnets, which then sends the pieces of the file to the reader.

The **WASTE** peer-to-peer system carefully controls which nodes may join the network. It is aimed at networks of 10-50 nodes and provides strong anonymity guarantees on condition that no one allows an attacker to join the network. Messages sent between nodes are encrypted, and idle nodes send dummy traffic, making traffic analysis attacks harder.

**Nodezilla** is an anonymous transport layer that uses the Everlink protocol. This protocol implements a version of Freenet in which both nodes and objects have identities, and requests are routed to nodes with the closest identity to the requested object. Unlike Freenet, when a node receives a packet and cannot forward that packet to another node, it assumes that it is the packet’s intended recipient. As with Freenet, nodes cache copies of the data sent across them. So, after a node forwards a file to its neighbour, the neighbour will know that the node is offering that file for download.

**GNUnet** is a searchable peer-to-peer file-sharing network whose transport protocol, called GAP is based on MIXes. In GAP, a user can choose to trade anonymity for efficiency by stopping the MIX nodes from rewriting the reply address of each message. GAP also uses an economic system where each request counts towards a node’s allocated credit. An analysis of GNUnet together with some possible attacks can be found in [Küg03].

**I2P** is a network layer that allows applications to communicate anonymously. I2P uses a technique called *garlic routing*, based on onion routing, in which the sender defines the path for outgoing messages and the recipient defines the path for incoming messages. Any of the intermediate nodes can inject a number of

hops before forwarding the message to the next peer. Another feature of I2P is that applications can select a tradeoff between latency and anonymity by adjusting some parameters of the protocol, such as the number of hops in their tunnels.

**AP3** [MOP<sup>+</sup>04] is a peer-to-peer overlay network which provides anonymous message delivery, anonymous channels and secure pseudonyms. It uses the same technique as Crowds but it is built on top of the Pastry [RD01] network. In Pastry random delivery is possible without knowledge of the whole network. A user can choose a random key and the network can deliver the message to the node which is “closest” to the key. AP3 does not have a built-in search, but as it supports anonymous multicasting one could be implemented.

**Tor** [DMS04] is an unsearchable transport layer system that uses Onion Routing. With care, it is possible to make standard file-sharing applications anonymous by running them over the Tor network. The Azureus BitTorrent client, for instance, provides the option of running over Tor. Responder anonymity is not part of the basic system, but it is made possible by “rendezvous points”.

Han et al. [HLX<sup>+</sup>05] have designed a file-sharing system based on Shamir’s secret sharing scheme called the **Secret-Sharing-based Mutual anonymity Protocol (SSMP)**. It requires a search to be broadcast to all nodes, and for some messages to be forwarded randomly until they reach their goal, so the system scales badly to larger networks.

There are also a number of other anonymous peer-to-peer systems such as **Share**, **Rodi** or **UDPP2P**. Unfortunately, they lack proper documentation, so it is hard to assess their merit.

## 4 Possible Attacks Against Anonymous P2P Networks

The attacker in a peer-to-peer system may be the sender, the responder, any node in the system or an outsider. The usual goal is to find out who the sender or responder is, or what they are transferring. Alternatively, we can consider a global attacker who can see the whole network, with the additional goal of linking the sender and receiver. Attacks can become much more effective if there are a number of attackers working together. If attackers can mostly surround a node, in any system, they can usually degrade that node’s anonymity. Knowledge of the network topology is also useful for an attacker. Just as many implemented systems combine a number of the basic methods for anonymity, real attacks on these systems may combine a number of the attack methods outlined below.

**Time-to-Live Attacks:** Time-to-live counters determine the maximum number of hops for a message and are used in most peer-to-peer networks to avoid flooding. If an attacker can send a request to a node with such a low time-to-live counter that the packet will probably not be forwarded, any response relieves that node as the responder. To avoid this problem, Mute, Ants, Freenet and Mantis use probabilistic time-to-live counters. In these systems there is always a chance that a request will be forwarded to another node.

Nodes in closely connected networks will receive copies of the same packet over each of its connections. If the attacker knows how the nodes are connected,

it may be able to work out the pseudo address of certain other nodes from the difference in the time-to-live counters of each of these copies.

There is also a time-to-live counter in the underlying Internet Protocol; In systems based on UDP spoofing, such as Mantis, the IP time-to-live counter could reveal some information about the sender.

**Multiple Attackers or Identities:** If attackers can make repeated connections to a single node with different identities, the anonymity of that node is usually lost. Some theoretical systems have complicated joining procedures to prevent attackers surrounding a single node. However, most implemented systems do not. This is one of the biggest security issues in up and running networks such as Mute or Ants. Attackers can make repeated connections to the same node and send that node a request for a file with an expired time-to-live counter. If the request is forwarded it will most likely go to one of the attackers, so if the attackers get any responses to their request they know it came from the node under attack. Douceur [Dou02] points out that it is almost always possible for a single attacker to assume a number of different identities, which he terms a “Sybil” attack. Another example of the use of multiple attackers is the locking or  $n - 1$  attack [GT96] on a MIX that mixes  $n$  messages. Here  $n - 1$  attackers send a messages to a MIX node. The next message sent to the node then triggers the MIX and can be traced, as it is the only message not known to the attacker.

One way to protect against this attack is to strictly control where a new node may join a network. A different approach is to have all nodes connected to all others, as in Crowds for instance. However, this tends to scale poorly to large networks. Even with these preconditions most systems will lose anonymity if the ratio of attackers to honest nodes gets too high.

**Statistical Attacks:** Any attacker can gather statistical data over time. Systems that are provably safe for a single run may reveal information about the identities of their participants when all the observable messages of a longer run are analysed for patterns.

An incorrect implementation of the Ants protocol search phase might call for a node to make a probabilistic choice to forward or drop a packet every time it receives one. While this would guarantee anonymity for a single run of the protocol, over time a node would be more likely to see requests with the identities of its direct neighbours. Mute defends against these kinds of attacks by fixing the probabilities of forwarding or dropping a packet when a node starts up. Hence, a repeated request will result in exactly the same action every time and provide no extra statistical information to the attacker. Another way to deal with this attack is to ensure that probabilities are used in such a way that all actions are equal likely. The nodes in Crowds forward messages to any other node in the system rather than just their nearest neighbours. When a node repeats an action, the message observable by the outsider is equally likely to come from any node.

If an attack can force a sender and a receiver to repeatedly re-establish a connection, or can identify the sender or receiver from the contents of a data stream, it may be able to gain the extra information it needs for a statistical



attack. This is the case in both Crowds and Onion Routing, as shown by Wright et al. [WALS01]. Shmatikov shows the same result for Crowds [Shm02] used the Prism model checker. Back, Möller and Stiglic [BMS01] describe a traffic analysis attack in the Freedom system. Raymond gives a good, but somewhat dated, review of this area [Ray00].

**Time-based attack:** The time a responder takes to respond to a request may indicate how far away the responder is from the attacker, or how many steps the system took. Especially when combined with a statistical attack, this can provide some information on the responder and help to compromise their anonymity. Adding in artificial delays will help to mask these information leaks but at the cost of performance (although, if implemented correctly the overall delay when downloading a large file can be negligible). Levine et al. have looked at time-based attacks and defences for Mix-based networks [LRWW04], much of which could also be applied to file-sharing systems. Serjantov and Sewell [SS03] look at time-based attacks for connection based systems.

**Attacks as Nodes Leave or Join:** If a node leaves a network it can no longer send a signal. So, in a system using pseudo identities, if a node leaves the network and the attacker still observes requests using a certain identifier, then the attacker knows that the node that left cannot have had that identity. As more nodes leave a network, the possible pool of nodes for a given long-lived identity will become smaller. In the same way, if an attacker sees  $n$  identities in a network with  $n$  nodes, then observes a node join and searches messages with a new identity, it is highly probable that the new identity belongs to the new node. The best defence against this sort of attack is for the nodes to regularly change their identities. As long as the life time of an identity is shorter than the average time a node is a member of the system, the system should be safe. Wright et al. [WALS03] describe passive logging and intersection attacks against onion routing when nodes leave and join the system.

**Denial of Service Attacks:** A peer-to-peer system cannot be used for anonymous downloading if it cannot be used at all. Denial of service (DoS) attacks can be particularly awkward when nodes can act anonymously, as this could mean that the node performing a DoS attack could not be identified and removed from the system. While anonymous systems cannot stop all DoS attacks, care should be taken to ensure that their design does not make DoS attacks particularly easy. Dumitriu et al. [DKK<sup>+</sup>05] have carried out an in depth study of DoS attacks in non-anonymous peer-to-peer networks. An attacker may carry out a DoS attack on another node in the Ants protocol by repeatedly broadcasting search messages using the other node's pseudo identity. If the attacker sends more search messages than the node under attack, all messages and files sent to the pseudo identity will be routed to the attacker. This could be avoided by using an authentication key as a pseudo address and including a signature of the message identity in the message. This way only the original user can generate new messages.

An attacker can gain a considerable advantage if it can use a targeted DoS attack to knock out a single node. The attacker could then observe the effect on communication of removing a node from the system, or it could remove a number of nodes in order to reshape the network.

## 5 Conclusion

The implementations described here are all in the early stage of development, so none of them provide the strong guarantees promised by the theoretical models. Currently, the advantage is on the side of the attacker. However, once the systems mature, and provided there is sufficient interaction between theory and implementation, real anonymity will be possible in a practical peer-to-peer file-sharing system.

Interest in the field of anonymity and anonymous peer-to-peer systems has grown rapidly. Papers on this subject are being published continuously; the Free Haven project keeps a useful list of these at <http://freehaven.net/anonbib>.

## References

- [BASM04] Steve Bono, Christopher A. Soghoian, and Fabian Monrose. Mantis: A high-performance, anonymity preserving, p2p network, 2004. Johns Hopkins University Information Security Institute Technical Report TR-2004-01-B-ISI-JHU.
- [BMS01] Adam Back, Ulf Möller, and Anton Stiglic. Traffic analysis attacks and trade-offs in anonymity providing systems. *Lecture Notes in Computer Science*, 2137:245–??, 2001.
- [Cha81] David Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 4(2), February 1981.
- [Cha88] David Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Communications of the ACM*, 24(2), 1988.
- [CSWH01] Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore W. Hong. Freenet: A distributed anonymous information storage and retrieval system. *Lecture Notes in Computer Science*, 2009:46+, 2001.
- [Dee89] Steve Deering. Rfc 1112 "host extensions for ip multicasting", August 1989.
- [DFM00] Roger Dingledine, Michael J. Freedman, and David Molnar. The free haven project: Distributed anonymous storage service. In *In Proceedings of the Workshop on Design Issues in Anonymity and Unobservability*, July 2000.
- [DKK<sup>+</sup>05] D. Dumitriu, E. Knightly, A. Kuzmanovic, I. Stoica, and W. Zwaenepoel. Denial-of-service resilience in peer-to-peer file sharing systems. *SIGMETRICS Perform. Eval. Rev.*, 33(1):38–49, 2005.
- [DMS04] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. In *Proceedings of the 13th USENIX Security Symposium*, 2004.
- [DO00] Shlomi Dolev and Rafail Ostrovsky. Xor-trees for efficient anonymous multicast and reception. *ACM Transactions on Information and System Security*, 3(2):63–84, May 2000.

- [Dou02] J. Douceur. The sybil attack, 2002. In Proceedings of the IPTPS02 Workshop, Cambridge, MA (USA), March 2002.
- [GSB02] Mesut Gunes, Udo Sorges, and Imed Bouazzi. Ara – the ant-colony based routing algorithm for manets. In *Proceedings of the International Workshop on Ad Hoc Networking (IWAHN 2002)*, Vancouver, August 2002.
- [GT96] Ceki Gulcu and Gene Tsudik. Mixing email with babel. In *SNDS '96: Proceedings of the 1996 Symposium on Network and Distributed System Security (SNDS '96)*, page 2. IEEE Computer Society, 1996.
- [HLX<sup>+</sup>05] Jinsong Han, Yunhao Liu, Li Xiao, Renyi Xiao, and Lionel M. Ni. A mutual anonymous peer-to-peer protocol design. In *ipdps, vol. 1, no. 1*, page 68. IEEE, 2005.
- [Küg03] Dennis Kügler. An analysis of gnutel and the implications for anonymous, censorship-resistant networks. In *Privacy Enhancing Technologies*, Lecture Notes in Computer Science, pages 161–176. Springer, 2003.
- [Ley04] John Leyden. Japanese p2p founder arrested. *The Register*, 10th May 2004. [http://www.theregister.co.uk/2004/05/10/winyou\\_founder\\_arrested/](http://www.theregister.co.uk/2004/05/10/winyou_founder_arrested/).
- [LRWW04] Brian N. Levine, Michael K. Reiter, Chenxi Wang, and Matthew K. Wright. Timing attacks in low-latency mix-based systems. In *the Proceedings of Financial Cryptography (FC '04)*, February 2004.
- [MOP<sup>+</sup>04] Alan Mislove, Gaurav Oberoi, Ansley Post, Charles Reis, Peter Druschel, and Dan Wallach. Ap3: A cooperative, decentralized service providing anonymous communication. In *Proceedings of the 11th ACM SIGOPS European Workshop*, Leuven, Belgium, September 2004.
- [PK04] Andreas Pfizmann and Marit Khentopp. Anonymity, unobservability, and pseudonymity: A proposal for terminology, draft v0.21, September 2004.
- [Ray00] Jean-François Raymond. Traffic Analysis: Protocols, Attacks, Design Issues, and Open Problems. In H. Federrath, editor, *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, pages 10–29. Springer-Verlag, LNCS 2009, July 2000.
- [RD01] Antony Rowstron and Peter Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, pages 329–350, November 2001.
- [RR98] M. Reiter and A. Rubin. Crowds: anonymity for web transactions. *ACM Transactions on Information and System Security*, 1(1):66–92, 1998.
- [SGR97] P F Syverson, D M Goldschlag, and M G Reed. Anonymous connections and onion routing. In *IEEE Symposium on Security and Privacy*, 1997.
- [Shm02] Vitaly Shmatikov. Probabilistic analysis of anonymity. In *IEEE Computer Security Foundations Workshop (CSFW)*, pages 119–128, 2002.
- [SLS01] V. Scarlata, B. Levine, and C. Shields. Responder anonymity and anonymous peer-to-peer file sharing, 2001. In Proceedings of IEEE International Conference on Network Protocols (ICNP) 2001.
- [SS03] Andrei Serjantov and Peter Sewell. Passive attack analysis for connection-based anonymity systems. In *Computer Security ESORICS 2003*. Springer-Verlag LNCS, October 2003.
- [WALS01] M. Wright, M. Adler, B. Levine, and C. Shields. An analysis of the degradation of anonymous protocols. Technical report, University of Massachusetts, Amherst., April 2001.
- [WALS03] M. Wright, M. Adler, B.N. Levine, and C. Shields. Defending anonymous communication against passive logging attacks. In *Proc. IEEE Symposium on Research in Security and Privacy*, Berkeley, CA, May 2003.

# A Churn-Resistant Strategy for a Highly Reliable P2P System

Giscard Wepiwé<sup>1</sup> and Sahin Albayrak<sup>2</sup>

<sup>1</sup> Technische Universität Berlin, DAI-Labor  
<http://www.dai-labor.de>

<sup>2</sup> Faculty of Electrical Engineering and Computer Science,  
Franklinstr. 28/29, D-10587 Berlin, Germany  
{[giscard.wepiwe](mailto:giscard.wepiwe), [sahin.albayrak](mailto:sahin.albayrak)}@dai-labor.de

**Abstract.** This paper proposes a churn-resistant strategy designed on top of a highly reliable P2P overlay network [1], with degree<sup>1</sup>  $2\Delta + 2$ , where  $\Delta$  is the degree of a De Bruijn digraph [2]. We show that when each node in the network periodically retransmits only one KEEPALIVE message to one of its neighbors in the network, any node's failure can be detected within an optimal timeout. As a major contribution, we demonstrate that even in failure situations the lookup of any available resource is achieved with the lowest possible maintenance overhead  $O(1)$  along the shortest path of length  $D_{CMR} = \log_{\Delta}(N(\Delta - 1) + \Delta) - 1$  with  $N$  being the maximal number of nodes in the network.

## 1 Introduction

The proliferation of Internet-scale services and the advent of peer-to-peer (P2P) applications for data sharing have brought about considerable attention to the resource distribution and lookup problem in dynamic distributed computer environments such as P2P systems. An important characteristic of P2P systems is the dynamic nature of their nodes, which are in a continuous process of “join and leave”. The result is a topology changing network where resource management issue is a highly challenging one.

Studies on dynamic graph theory [3] have shown that low-degree networks suffer from network disconnection when even only one node is deleted. On the other hand, networks of large degrees are indeed robust but susceptible to collapse as some nodes collectively fail. Moreover, it is a fact that fast detection of failures enables rapid recovery and contributes to keep the network in a stable state. This means that the frequent retransmission of periodic information to the neighborhood can significantly reduce the failure detection time. However, frequent retransmission of presence information (also called KEEPALIVE message in this paper) at high rate to neighbors would increase control overhead

---

<sup>1</sup> The degree of a vertex  $\Delta(x)$  is the number of vertices adjacent to  $x$ . The degree of a graph  $G$  is  $\Delta_G = \max\{\Delta(x), x \in V\}$ . With  $G$  defined as  $G = (V, E)$ , where  $V$  is the set of vertex and  $E$  is the set of Edge of  $G$ .

in the network. Although several P2P proposals for churn management have been proposed recently to deal with these issues [4, 5, 6, 7, 8], many fundamental questions remain unanswered. Some of these are:

- How to conceive network maintenance strategies such that the deletion of a node or a subset of nodes does not cause the network to become disconnected or to collapse.
- How to choose the timeout for failure detection such that (i) the susceptibility to use the alternative routing solution is kept low and (ii) the control overhead and the message lost are minimized.

In what follows, we propose a churn-resistant strategy which responds to these open questions. The strategy is to be deployed on top of a highly reliable P2P network overlay, which is presented briefly in the Section 2. Interested readers may refer to [1] for more details.

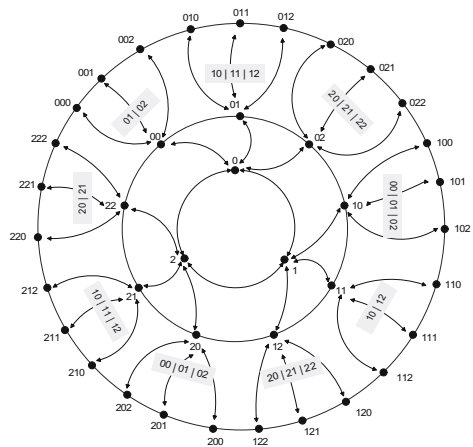
## 2 Overview of the Concentric Multi-ring Overlay (CMR)

The churn-resistant strategy presented in this paper is designed to detect and recover failure in a P2P network based on the CMR overlay. For convenience, we give an overview of the CMR network model in this section.

### 2.1 System Model

A CMR-based network is organized as a concentric multi-ring representation, where each of the  $N$  nodes —  $N$  as the total number of nodes in the system — are interconnected with  $2\Delta+2$  neighbors as shown in Figure 1. The CMR representation consists of several concentric rings, numbered with ring identifier (RID) from 1 for the innermost ring to  $D_{max}$  for the outermost ring. Nodes on a ring with identifier  $D$  are interconnected so to form a De Bruijn graph [2] with diameter  $2 D$  and degree  $\Delta$ . Each node in the network is assigned an identifier (ID) and each ring consists of  $\Delta^D$  nodes. The node's ID is a  $\Delta$ -base number with  $D$  digits; e.g the node  $x=213121$  is a 4-base integer located on the ring with identifier 6.

According to the dynamic behavior of the P2P systems, the network can grow as necessary. Nodes always join



**Fig. 1.** The Logical Representation of Nodes on the Concentric Multi-ring Overlay

<sup>2</sup> Diameter of a graph is defined as the minimum distance between the two most distant nodes in the network.

the overlay network on its outermost ring. The ring's construction always begins with the node  $x_{\{init,D\}}$  having the lowest identifier on the ring; where  $x_{\{init,D\}} = x_{D-1}x_{D-2}\dots x_1x_0$ , with  $x_i = 0$  and  $i \in \{0, 1, \dots, D - 1\}$ . The total number of nodes in the network is at most  $N_{max} = \frac{\Delta^{D+1} - \Delta}{\Delta - 1}$  with  $\Delta > 1$ .

In this work, we choose to interconnect nodes on the same ring after the De Bruijn principles owing to two reasons: (i) nodes in a De Bruijn network require no more than  $\Delta$  neighbors to ensure routing in the shortest path; (ii) the routing distance between any two peers in a De Bruijn network is not longer than  $\log_{\Delta} N$  which is closer to the Moore bound [9]<sup>3</sup> than many other solutions based on Distributed Hash Table like Chord ( $\log N$ ) for example. However, when nodes join and leave continually in the system, there is a need to design management strategy in order to resist to churn and continue to route along the shortest path. The questions on how a failure is detected and how it is recovered, are detailed later on in Section 3.

### 2.2 Routing Scheme

Routing in our CMR network is an extension of the De Bruijn routing, where each node can forward a message toward its neighbor on the way to the destination. Each node  $x$  in a CMR ring with identifier  $D$  has at most  $2\Delta + 2$  neighbors, where messages can be routed depending on the destination ID; these are (i) the  $\Delta$  nodes  $y = x_{D-1}x_{D-2}\dots x_1\beta$  on the same ring; (ii) the  $\Delta$  nodes  $y = x_{D-1}x_{D-2}\dots x_1x_0\beta$  on the next outer ring; (iii) the node  $y = x_{D-1}x_{D-2}\dots x_1$  on the next inner ring; and (iv) the head node  $y = 0x_{D-1}x_{D-2}\dots x_1$  or  $y = 0x_{D-1}x_{D-2}\dots x_2$  if  $x$  is the ring contact node, with  $0 \leq \beta \leq \Delta - 1$ .

Each node  $x$  in the network maintains at most 2 tables:

- A routing table with the List\_1 of pairs (*NodeID*, *IP Address*) for maintaining information about the De Bruijn neighbors and the List\_2 of pairs (*NodeID*, *IP Address*) for maintaining information about the outer ring, inner ring neighbors and the head node.
- A resource location table with the list of pairs (*GUIDE*, *Provider's IP Address*). Each resource is represented in the network with a Global Unique Identifier (GUIDE) which is a  $\Delta$ -base integer.

To route between any two peers  $x$  and  $y$  in the network, a next-hop is chosen on-demand at each step on the way to the destination. Results in [1] show that routing between any two nodes in the network can be achieved within at most  $D_{CMR} + 1 = \log_{\Delta}(N(\Delta - 1) + \Delta)$  overlay hops.

### 2.3 Resource Distribution Scheme

The resource distribution scheme specifies how resources are distributed among the peers in the network. Each node  $x$  on ring with ID  $D_x$  can place information

<sup>3</sup>  $\log_{\Delta}(N(\Delta - 2) + 2) - \log_{\Delta-1} \Delta$  with  $\Delta > 2$ .

about its resource  $R$  of length<sup>4</sup>  $L$  at any node  $y = R \div \Delta^{L-D_x}$  on the same logical ring with ID  $D_x$  and at any node  $y = R \div \Delta^{L-D_x+1}$  on the next inner ring<sup>5</sup> with identifier  $D_x - 1$ . As the network grows, the resource placement is extended to all nodes  $y = R \div \Delta^{L-D_x-i}$ , with  $i \in \{1, 2, \dots, D_{max} - D_x\}$  on the outer rings. In this paper, node  $y$  indicates the ambassador of  $x$  for the resource  $R$  in the network.

**Lemma 1.** *For any resource  $R$ , each node in the network has at least 1 and at most  $D_{max}$  ambassadors in the network. Where  $N \geq \Delta$  and  $D_{max}$  being the total number of rings forming the network.*

*Proof.*  $R \div \Delta^{L-D_x}$  is the ID of exactly one node on the same ring as  $x$ , when  $N \geq \Delta$ . As the network grows, the outer rings are constructed and there is at most one more node with ID equal to  $R \div \Delta^{L-D_x-i}$  when  $0 < i < D_{max} - D_x$ . Thus, in a network with a total of  $D_{max}$  rings, each node  $x$  has an ambassador for its resource  $R$  on all the  $D_{max} - D_x + 1$  rings with ring identifier  $A$ , so that  $A \geq D_x - 1$ .

Hence, each node  $x$  has at most  $D_{max}$  ambassadors in the network.

### 2.4 Resource Lookup Scheme

As resources are always distributed from the inner to the outer ring, a node  $z$  on ring  $D_z$  looking for a resource  $R$  of length  $L = |R|$  provided by  $x$  sends a resource lookup request to the node  $\bar{x} = R \div \Delta^{L-D_z}$ . With a high probability, node  $\bar{x}$  knows  $x$ , a provider of  $R$ . The node  $\bar{x}$  and the node on the way to  $x$  forwards the message to their outer ring until a possible node  $y$  is reached that knows the provider of the resource  $R$  or until the provider itself is reached.

**Theorem 1.** *Resource lookup in the network can find an ambassador of the resource provider in at most  $D_{CMR}$  hops.*

*Proof.* If a resource  $R$  exists in the network, its furthest provider can be located after at most  $D_{CMR} + 1$  forwarding hops as stipulated in Section 2.2. If a provider is located on the outermost ring with identifier  $D_{CMR}$ , then according to Lemma 1 there is a node  $y$  on the ring with identifier  $D_{CMR} - 1$  which is ambassador for  $R$ . To route from any ring in the network to the ring with identifier  $D_{CMR} - 1$ , one needs at most  $D_{CMR}$  hops. In conclusion, a node  $y = R \div \Delta^{L-D_{CMR}+1}$  is reached after at most  $D_{CMR}$  hops, which has a location table entry with the resource  $R$ .

## 3 Network Maintenance

A CMR-based network is in a consistent state if and only if all the inner rings are fully constructed and each node on the outermost ring knows its head node.

<sup>4</sup> It is assumed that  $L > D$ .

<sup>5</sup>  $\div$  is used for the division of two  $\Delta$ -base integers.

The head node of a node  $x$  is the node to which  $x$  regularly sends KEEPALIVE message. In this manner, the nodes in the network must be able to detect node’s failure in their neighborhood in order to start recovery procedure and bring back the network to a consistent state.

### 3.1 Node Arrival

To join the network, a node must know the IP address of at least one node already in the network — the so-called “network contact node”. In the following description, we assume that a node  $x$  willing to join the network is aware of at most one network contact node. In order to join however, the node  $x$  should: (i) determine the identifier of the outermost ring and (ii) get a node’s ID from the network.

*Determine the actual outermost ring:* A joining node  $x$  sends a “join request” message to a network contact node, asking for the attribution of an ID on the actual outermost ring. The “join request” message is routed to the node  $x_{\{init,D\}}$  on the outermost ring, which ensures that an ID is assigned to  $x$ . If the outermost ring is full, then a new outer ring is created and  $x$  becomes the node  $x_{\{init,D+1\}}$  on that ring, else the newcomer is attributed an identifier in the middle of the next larger space on the outermost ring. The operation of determining the actual outermost ring may involve at most  $D$  nodes to forward the “join request” message.

*Determine the ID of a node joining the network:* To better illustrate the node joining procedure, let’s have a look of the abstract ring representation in form of a tree in Figure 2.

To ensure an even distribution of keys at all times in the system, the outermost ring’s contact node has  $\Delta - 1$  branches with well-defined branch size. The branch size

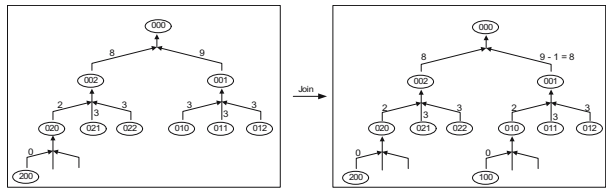


Fig. 2. Tree representation of the ring with  $RID = 3$

decreases as a new node is inserted on the ring. The original size of a branch is computed using the formula  $branchsize = \frac{\Delta^{D-index} - 1}{\Delta - 1}$  with  $index$  being the position of the first occurrence of a non-zero digit from the left of the node’s ID; for example the original size of the 3 branches of the node 020 is 1. After each “join request” message, the outermost ring’s contact node forwards the message to the branch that has the highest branch size and decreases the related branch size by one. A new ring is created, if all the branches of the ring contact node have a size being equal to zero. The head node is the node who sent the “join invite” message to the newcomer. Hence, after joining the system the new node notifies its presence to its neighbors, distributes its resources, and



collects notification messages received from others nodes to construct its tables. After this, the node is connected and can request and exchange resources from respectively with the rest of the network.

Each node joining the network generates one message to determine the ID of outermost ring, one resource location message to distribute each of its resources, and at most  $\Delta + 1$  messages to update its routing table. This makes a total number of  $\Delta + 3$  messages involving at most  $D$  nodes for forwarding.

### 3.2 Failure Detection and Recovery

In order to keep the network in a consistent state, the failures of nodes due to e.g. sudden or intentional departures should be detected, and their effect on message routing and resource lookup operations in the network should be minimized.

*Network Failure Detection:* In order to still aware of the presence of their neighbors, any node  $x$  on the logical overlay network informs after a time period  $T$  its head node denoted as  $x^{head}$  about its presence in the network. The choice of  $T$  is conditioned by the dynamic pattern of the network. In this work, we consider two cases of node departure in the network. When a node  $x$  voluntary leaves the network, it informs a successor about its intention to leave. However, when a node has been deleted from the network by a malicious actor for example, its head neighbor must automatically detect its absence after a given timeout. That is, when the head neighbor of a node  $x$  does not receive any message from  $x$  after a timeout period  $T_{out} = f(T)$ , then  $x$  has probably crashed. The study presented in [4] demonstrates that the choice of the timeout value  $T_{out}$  is a significant factor for the network reliability and hence for the lookup latency under churn. For our network simulation, we choose the style of timeout calculation as in mature TCP works such as [10] for message retransmission during network congestion. Hence, we choose

$$T_{out} = T + \frac{AVG}{2} + 4 \times VAR,$$

where  $AVG$  is the observed average round-trip time and  $VAR$  is the observed mean variance of that time;  $T$  is derived from experimental network measurement study of a KaZAa network [11]. That is, after each period  $T_{out}$ , if the head neighbor does not receive any message from  $x$ , then the node is assumed as probably failed. We say probably, because it may happen that  $x$  is still available in the network but the link between  $x$  and  $x^{head}$  is congested. In this work, we assume that the absence of a KEEPALIVE message between two nodes  $x$  and  $x^{head}$  is a sign that  $x$  has failed.

*The Network Recovery Procedure:* Once a node  $x^{head}$  has detected the failure of a node  $x$ , two operations are processed. First, node  $x^{head}$  sends a “route obsolete message” to all the nodes, who know  $x$  as a forwarding hop. On receipt of a “route obsolete message”, nodes update their routing table. Secondly, node  $x^{head}$  chooses a successor for the node  $x$ . Therefore, it sends a “choose

successor” message to the outermost ring’s contact node, which forwards it to the smallest leaf on the tree representation of the ring. The smallest leaf leaves the outermost ring and replaces the failed node on the inner ring. Once on the new ring with a new ID, the node subsequently informs its ongoing De Bruijn neighbors on the same ring. It also informs its inner ring’s neighbor and its  $\Delta$  neighbors on the next outer ring about its new IP address. This stabilization process enables the network to converge very fast and generates only  $2\Delta + 2$  messages.

Further, the new node  $x$  chooses one ambassador for each of its resource.

### 3.3 Concurrent JOIN and LEAVE

We also study the complexity of concurrent join and leave events in the network.

The concurrent JOIN protocol is similar to the isolated JOIN protocol with a difference on the join duration delay. That is, when  $k$  nodes want to join the network simultaneously, their requests are processed subsequently and the join request can be delayed.

The concurrent LEAVE protocol is a challenging issue which basically relies on the isolated LEAVE protocol. After a malicious attack, several nodes on different rings could detect failure of their neighbor. In any case, the failure detection is done after a time  $t$ , with  $0 \leq t \leq Tout$ . The recovery from a concurrent LEAVE event is a successive processing of the isolated network recovery procedure.

Additionally, if a request reaches a deficient node before the recovery from all failures is complete, the adaptive routing scheme is applied as described in Subsection 3.4.

### 3.4 Static Resilience Scheme

The routing between any two nodes  $x$  and  $y$  in the network is a multi-hop forwarding operation as shown in Section 2.2. However, the routing operation presented there assumes that the network is in a consistent state and will consequently fail to route packets to its destination when a neighbor  $z$  along the path from  $x$  to  $y$  suddenly fails. To address this issue, we propose a static resilience scheme which enables routing around the failure region.

When the computed next-hop node  $z$  is not available in the table, the message is forwarded to the alternative node  $\bar{x}$  which is chosen as follows:

- If the message target  $y$  is a node on an outer ring and node  $z$  is on the same ring than  $x$ , then  $\bar{x}$  is the neighbor of  $x$  on the next outer ring such that  $\bar{x}$  is the node closest to the node  $prefix(D_{\bar{x}}, y)$  in the ID-space<sup>6</sup>.
- Else  $\bar{x}$  is the head node

**Theorem 2.** *Any resource request in the network is forwarded along the shortest path to its destination, even at high rate of node failure.*

<sup>6</sup>  $prefix(i, y)$  returns the first  $i$  letters to the left of the word  $y$ .

*Proof.* The static resilience scheme in this subsection ensures that a packet can be routed around a failure region. Here, we choose the most optimal next-hop for packet forwarding so that the routing operation as specified in Section 2.2 can still be applied. Additionally, we've shown that routing in the network is always achieved along the shortest path in the consistent network state. Thus, we can conclude that any request in the network is always forwarded along the shortest path to its destination even in failure situations.

## 4 Performance Evaluation

In this section, we evaluate the CMR protocol by means of a network emulation. The performance evaluation is given for high churn rate situation with little or no congestion. Our findings are based on nearly real network conditions as studied in [11]. We emulate a network of about 1300 nodes running on 7 SUN Fire V210 machines with 2x1,0GHz UltraSPARC III 64 Bits with 2GB RAM; all the machines are connected with each other over a Gigabit Ethernet. We simulate node joining and leaving the network, resource distribution and resource request. During the simulation, one node is inserted in the network each two seconds. After the first five nodes have been initialized in the network, we start the churn situation at the rate of one failure or one leave after each five joins. While the network is growing, each node sends one KEEPALIVE message to its head node at a rate of one per 10 seconds and requests one arbitrary selected resource every 15 seconds.

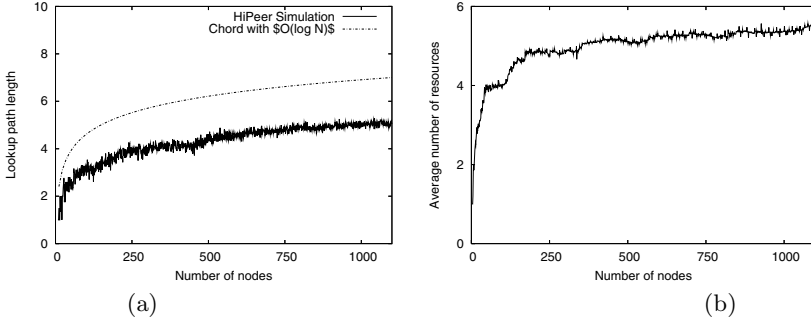
In the sequel, we discuss some measurements results.

*Lookup Path Length:* At high rates of churn at least 99.5 percent of the 194109 requested resources are found by the requesters. Figure 3(a) shows that each successful lookup is achieved within  $D_{CMR} = \log_{\Delta}(N(\Delta - 1) + \Delta) - 1$  hops and compares the average path length of CMR against Chord [5]; this result reveals that lookup in CMR is achieved within a lower path length than Chord at high rate of churn.

*Load Balancing:* We show in Figure 3(b) that even when many nodes come and go continually the available resources are fairly distributed between nodes in the network. The Figure 3(b) shows that the average distribution of resources per node is almost constant independently of the network size.

*Routing Table Size:* Figure 4(a) shows the evolution of the mean routing table size per node in a network. The routing table size is limited to maximum  $2\Delta + 2$ .

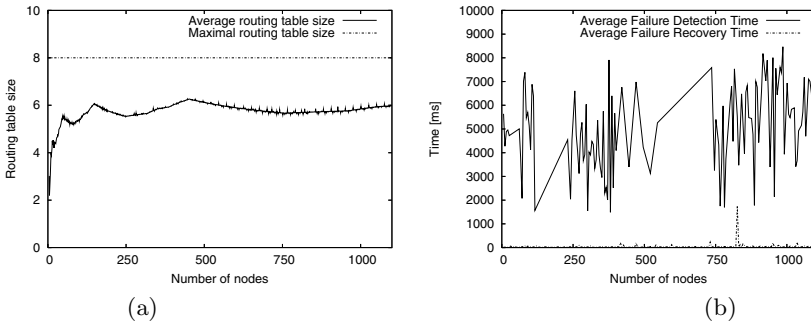
*Failure Detection Delay and Recovery Delay:* For our emulation, we set  $T = 10$  seconds for sending regular KEEPALIVE message. The graph of Figure 4(b) shows that the failure detection time mainly depends on  $T$ . The graph of Figure 4(b) also presents the failure recovery delay, which is the time spent from the failure detection time until the recovery operation is complete. We can see that the failure recovery time is very short, depending on whether the failed node and the successor are on the same emulation machine or not.



**Fig. 3.** (a) The average lookup path length of CMR with degree  $\Delta = 3$  compared to Chord. (b) The average number of resources stored per node in a failure-prone growing network.

### 5 Related Work

A plethora of different overlay networks with various interesting technical properties have been developed over the last five years [5, 8]. The protocols not only aim at defining scalable overlay structure for fast lookup of resources in P2P network environments but also address the failure resilience issue. In order to support resilience, they are mostly based on two principal classes of failure detection algorithms: (i) the reactive approach, where the nodes only send KEEPALIVE messages in data packets and (ii) the proactive approach, where the nodes periodically send KEEPALIVE messages to their neighbors. The reactive approach is generally used in the literature as an optimization of the active approach. Thus, many proposals for P2P network resilience techniques such as [4, 6] rely on KEEPALIVE techniques to detect failure of peers in the network. The study of failure detection algorithms in [7] shows that the delay during failure detection is an important performance factor for resilience in P2P systems; thus, it is



**Fig. 4.** (a)- The average failure detection and recovery time in a highly rate of network “join and leave”. (b)- The average number of routing table entries per node as a function of the network size.

worth to minimize the detection time of a node's failure. Moreover, in order to minimize the delay until the detection of a failure in the network further, some timeout calculation techniques are presented in [4].

## 6 Conclusion

We have presented a churn-resistant algorithm designed on top of a highly reliable concentric multi-ring overlay for management of large-scale highly dynamic P2P systems. We have proposed a network maintenance strategy where each node periodically retransmits only one control message to remain connected. A major contribution of the strategy is its ability to enable network consistency and resource lookup in at most  $D_{CMR} = O(\log_{\Delta}(N(\Delta - 1) + \Delta) - 1)$  overlay hops even at high rate of churn with a lookup success percentage of about 99, 5.

## References

1. Giscard Wepiwé and Plamen L. Simeonov. A concentric multi-ring overlay for highly reliable p2p systems. In *Network Computing and Applications (NCA05)*, Cambridge, MA, USA, July 2005. IEEE Computer Society. To Appear.
2. N.G. de Bruijn. A combinatorial problem. In *Nederl. Akad. Wetensh. Proc. 49*, pages 768–764, 1946.
3. Ding-Zhu Du and D. Frank Hsu. *Combinatorial Network Theory*, volume 1 of *1948-III Series*, chapter 3: De Bruijn Digraphs, Kautz Digraphs, and Their Generalizations, pages 65–105. Kluwer Academic Publishers. Printed in the Netherlands, 1996.
4. Sean Rhea, Dennis Geels, Timothy Roscoe, and John Kubiatowicz. Handling churn in a DHT. In *Proceedings of the 2004 USENIX Annual Technical Conference (USENIX '04)*, Boston, Massachusetts, June 2004.
5. Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. *SIGCOMM Comput. Commun. Rev.*, 31(4):149–160, 2001.
6. Simon S. Lam and Huaiyu Liu. Failure recovery for structured p2p networks: protocol design and performance evaluation. *SIGMETRICS Perform. Eval. Rev.*, 32(1):199–210, 2004.
7. Shelley Zhuang, Dennis Geels, Ion Stoica, and Randy Katz. On failure detection algorithms in overlay networks. In *INFOCOM'05*, 2005. to appear.
8. Ben Y. Zhao, Ling Huang, Sean C. Rhea, Jeremy Stribling, Anthony D Joseph, and John D. Kubiatowicz. Tapestry: A global-scale overlay for rapid service deployment. *IEEE J-SAC*, 22(1):41–53, January 2004.
9. Biggs Norman L. *Algebraic Graph Theory*. Cambridge Math. Library, 1974, 1993 (2nd edition), 1993.
10. V. Paxson, M. Allman. Computing TCP's Retransmission Timer. Request for Comments: 2988, November 2000.
11. Krishna P. Gummadi, Richard J. Dunn, Stefan Saroiu, Steven D. Gribble, Henry M. Levy, and John Zahorjan. Measurement, modeling, and analysis of a peer-to-peer file-sharing workload. In *SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles*, pages 314–329, New York, NY, USA, 2003. ACM Press.

# Building a Peer-to-Peer Overlay for Efficient Routing and Low Maintenance

Honghao Wang and Yiming Hu

University of Cincinnati, Cincinnati, OH, 45219, USA

**Abstract.** Most of current structured P2P systems exploit Distributed Hash Table (DHT) to archive an administration-free, fault tolerant overlay network and guarantee to deliver a message to its destination within  $O(\log N)$  hops. While elegant from a theoretical perspective, those systems face difficulties in a realistic environment. Instead of building P2P overlays from a theoretical perspective, this design tries to construct an overlay from the physical network. By combining different network topology aware techniques, a distinctive overlay structure closely matching the Internet topology is created. The P2P overlay based on this structure is not only highly efficient for routing, but also keeps maintenance overhead very low even under highly dynamic environment.

## 1 Introduction

Most of current structured P2P overlays exploit Distributed Hash Table (DHT) to archive an administration-free, fault tolerant overlay network and guarantee to deliver a message to its destination within  $O(\log N)$  hops.

While elegant from a theoretical perspective, DHT-based systems face difficulties in routing efficiency and high overhead in a realistic environment. Although DHT designs guarantee to solve a query within  $O(\log N)$  hops, the previous study [1] has shown that a significant fraction of nodes could be connected over high latency / low bandwidth links. The presence of even one such slow logical hop on a logarithmically long path is thus likely. This increases the overall cost of the lookup. Recent DHT designs have tried to optimize routing by exploiting physical network information, however, latest research [2] has pointed out that the latency of last few hops in a lookup still approximated 1.5 times the average round trip time under Proximity Neighbor Selection (PNS). The situation will be worse under dynamic environment. As mentioned by Rhea et al. in [3], for 1000 nodes system under modest churn rate, a Pastry system (FreePastry) failed to complete 70% requests. While almost all lookups in a Chord system were completed, the lookup latency increased more than 20 times.

In order to handle system churn, the node in the latest Pastry, Bamboo DHTChurn system is designed to periodically detect its neighbors and share the leaf set with them. The overhead of those operations is substantial. The bandwidth is about 1.8kps per node, and the traffic is more than 1.5TBytes daily for a 100,000 system, which is 7.5 times larger than the Gnutella system [4].

In order to build an efficient structured P2P overlay in a realistic environment, this paper, from a different angle, proposes a new protocol. Opposite to current P2P designs,

our approach focuses on building the system overlay closely matching the physical network. As a result, physical network characteristics, such as the power law of the Internet, network locality among nodes and asymmetric throughput of major network connections, are fully exploited. By a novel piggyback mechanism, the system subtly integrates maintenance work into a common operation. As a result, the overhead of the system maintenance is reduced to the minimum.

The rest of this paper is structured as follows. Section 2 presents the background of the Internet topology and related techniques. Section 3 describes in detail how to construct the system overlay closely approximating the Internet topology and build an highly efficient system on it. In section 4, we evaluate this approach using simulation. Section 5 compares it to related work. Section 6 concludes the paper.

## 2 Internet Structure and Related Techniques

The Internet is made up of many Autonomous Systems (ASes). An AS, normally an ISP or organization like companies or universities, is a network under a single administration authority using a common Interior Gateway Protocol (IGP) for packet routing. Machines within an AS are normally geographically close and connected by LAN/MAN techniques. Some border routers running Border Gateway Protocol (BGP) connect it with neighboring ones. There are many methods and techniques to acquire network aware information, such as BGP routing tables [5, 6], widely used landmark technique [7] and the network of physical springs [8].

BGP routing tables can provide router-level Internet topology. However, they are not only hard to obtained, but also too complex to use. For AS-level information, lots of public services, such as the CIDR Report [6] and WHOIS service, are available. All information, such as IP address span, AS number and connectivity, are announced. Some service even updates daily and provides thorough analysis. By measuring RTTs between particular node and other nodes or pre-selected hosts (landmarks), landmark techniques and the Vivaldi can provide some kind of coordinates to reflect node's relative position in the Internet. Since many unpredictable factors, such as changes of routing, network bandwidth and traffic, can affect RTTs, those techniques are not that accurate and stable, and do not directly reflect network topology.

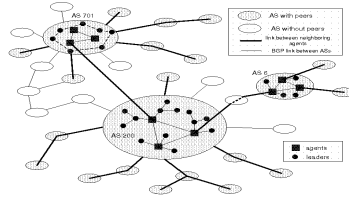
Above techniques will be used together in our design to ensure the overlay structure close matching the physical network, which will be discussed later.

## 3 Overlay Design

In this section, we will firstly discuss how to construct the overlay closely approximating the physical network, and then to build an efficient P2P overlay based on it.

### 3.1 Building an Overlay Based on Internet Structure

As Figure 1 shows, based on published AS-level Internet topology, all nodes are divided into *groups* by their AS locus. The group is the basic unit for routing and organizing nodes in the system. However, to partition nodes only by the AS may be too coarse,



**Fig. 1.** Building the system overlay closely matching Internet topology

especially for large ASes with lots of nodes. Thus, we propose to use landmark technique to further divide nodes into *teams*. The default routers of previously joined nodes would be good candidates for landmarks. As Ratnasamy et al. mentioned in [7], 6 to 8 random selected landmarks were enough to divide a network with thousands of hosts. Hosts will be grouped by their latency ordering of landmarks, which is called *landmark vectors*.

Several nodes with high bandwidth and availability will be elected within each group as *agents* to perform routing for the group. A leader will also be selected to route for the team. Although each peer arbitrarily joins and leaves the system, previous researches [1, 4] have showed that their behaviors were highly skewed. Most of nodes have very short uptime, however, there are 18% and 10% nodes with 90% uptime in Napster and Gnutella, respectively. Also, those 10% hosts also contribute 90% of the total traffic. Thus, those 10% nodes are appropriate candidates for agents and leaders. The more detail will be addressed in later sections.

### 3.2 Peer-to-Peer Overlay Design

In order to support ID lookup operation like structure P2P overlays, the similar ID mechanism is employed. Each object in our overlay has a 128bits ID, which can be generated by a basic hash function such as SHA-1. Instead of mapping a small range of objects to each node, a two levels mapping mechanism is used in our overlay. The first level is among groups. The second level is among teams. Since valid Internet AS number is from 1 to 64511 and only about 17,000 AS is active currently, a 32bits ID for groups is considered enough. Also, each team will be assigned a 32bits team ID. Each group and its teams will be assigned the unique ID randomly when they first appear. Each group will store the objects with the first 32bits between its group ID and the next one, which is similar to Chord [9]. Within a group, the object will be further mapped into a team by its second 32bits. In other words, given a 128bits object ID, the first 32bits is used to locate its responsible group, and the second 32bits is used to find the related team.

The team is the basic unit to store objects. Two copies of objects will be kept within a team for improved reliability and availability. One copy will be kept in the leader to reply queries for other peers. The other one will be striped into blocks by erasure code technique and store among teammates. Previous researche [2] has pointed out that erasure code technique can significantly improve dependability, and reduce bandwidth for updating objects. The only drawback is the read latency. However, our design successfully solves this problem. Since more than 50% peers are connected with asymmetric



network connections, such as DSL and cable modems, to retrieval information from several nearby peers will be evidently faster than from one. Also, this characteristic is helpful to quickly recover failure leaders and agents. The number of nodes within a team will be varied to facilitate clustering physical nearby nodes and keep team stable under extremely churn. Our experiments show that the suitable team size can be from 10 to 50 nodes.

**Routing.** The routing mechanism is actually simple compared with DHT designs. As mentioned early, since a two levels mapping mechanism is used in the overlay, the routing table is made up of two tables. One records each group's ID and information of agents within the overlay, called *routing table*. This table is maintained in each agent and leader to route messages for normal nodes. The other one is only kept in each agent for its group. It records each team's ID and leader's information within the group, called *delivering table*. Given an object ID, the responsible group and agents can be located by the routing table. Instead of hop by hop approaching the destination, messages will go directly to the agent in the target group. When the message reaches the agent, it simply forward that message to the response team leader by the delivering table. Finally, the leader replies the message. Normally, 3 agents can provide enough availability and performance without loading the host machines. A detailed analysis is given in section 3.3.

**Node Joining and Leaving.** The procedure of nodes' joining and leaving is simple. When a node  $N$  joins the system, its AS number can be determined by its IP by published services. The bootstrap protocol is straightforward. By any node within the system, the joining request will be forwarded to an agent of the node  $N$ 's AS. After measuring  $N$ 's landmark vector, the node will join one team according to its network locality. The overhead is minimal, since only the leader has to update some book-keeping information. If a team is too populous, it will split into two based on network locality. If the joining node is the first one in the AS, that request will be sent to a physical nearby group. Instead of forming a new group, the node will become a teammate within that group, called *mother group*. When nodes within that AS are enough for three teams, agents will be selected and an individual group is born.

For the normal node, its leaving or failure is automatically tolerated by the team. When an agent or a leader is leaving the system, a new one will be selected. As mentioned early, datum from different nodes can rebuild the routing tables and objects for the new one in seconds. Its information will quickly spread out by a piggyback mechanism, which will be discussed in the following section.

### 3.3 Maintenance

In P2P environment, not only each peer's leaving and failure is unpredictable, but also the whole system is highly dynamic. Thus, to build a system of efficiency and low maintenance under such dynamic environment is really a challenge for all P2P overlays. Although current DHT designs can tolerant nodes' failure, latest research [3] has showed that those systems degenerated quickly under dynamic environment.

Although agents and leaders are considered to have higher availability than normal nodes, they are not well-maintained servers. The leaving of agents and leaders will not

**Table 1.** All traffic generated by different roles and system functions. \* is the ratio to DSL or cable modem connection with 3Mb downlink and 384kb uplink.

System Operations	Agents (Bps)		Leaders (Bps)		Nodes (Bps)	
	Downlink	Uplink	Downlink	Uplink	Downlink	Uplink
Lookups	666.67	666.67	200	200	10	10
Ring Protocol	20	20	26.67	20	0	0.33
Update Others	44	264	0	0	0	0
Piggyback	400	0	0	120	0	0
Total	1131 (0.38%*)	951 (2.48%*)	271 (0.09%*)	560 (1.46%*)	10	10.33

impact the system much, since their datum can be rebuilt quickly. Although node's crash is not considered a common case, it should be quickly detected and recovered. A modified ring protocol [10] is used to monitor agents and leaders. As the AS 701 showed in Figure 1, all leaders will form a ring as their ID relationship. Every second, each node will send a keep-alive message to its successor and predecessor. By combining reports from other leaders, the agent can accurately find the crash leader. The same method is also used between agents. Normally, the leader can monitor teammates' changing through their query messages.

Since the leaders cache the routing table from group agents, some method is needed to keep them consistence. For small groups, like the AS6 in Figure 1, leaders can directly exchange information with a nearby agent. However, this simple schedule is not suitable for large AS. Since nodes are organized as the AS, a same network administration, administratively Scoped IP Multicast can be exploited. For the AS without multicast, *dissemination trees* with agents as the root will be used to multicast the information to all leaders, as the structure of AS200 in Figure 1 shows. For even larger AS, it will be partitioned into two or more groups.

To keep those routing tables up to date is critically important for lookup correctness. Also the overhead to maintain them should be low. Otherwise, both system performance and scalability will be affected. Thus, our overlay is designed to integrate this work into the common operation, *lookup*. When the leaders send out or answer messages, the information of changed agents, which is about 4 bytes for one agent, will be appended to the messages and reach an agent within the destination group, and then spread to all leaders. After that, the information will be further sent to more groups with outing messages. Actually, this gossip-style piggyback mechanism is highly efficient and robust. It is well known that with high probability all groups will be informed within  $O(\log N)$  turns, in which  $N$  is the number of groups. Assuming one turn costs about 3 seconds, which is the time for an update to reach and spread in a group, a system with 5000 groups for one million nodes could be updated within 4 turns, 12 seconds. Our experiments show that in a system with 10,000 nodes distributed in 100 ASes, assuming a median session time of 15 minutes for each node, the success rate of first attempt with the piggyback technique is 99.53%.

**Scalability Analysis.** Currently, the hosts of real world systems are distributed in 4 to 5.5 thousands ASes. The total number of nodes is from 200,000 to 1,000,000, and the number per AS varies from several to thousands [4]. Thus, the size of routing table with

5000 groups and their agents' information is about 60KB, and size of delivering table with 300 teams is about 3KB.

Previous research [3] has pointed out nodes' joining and leaving in the P2P environment can be modeled by a Poisson process. Thus, an event rate  $\lambda$  corresponds to a median inter-event period of  $\ln 2\lambda$ . Therefore a churn rate of  $\lambda$  corresponds of a median node session time of  $N$  nodes within a network is the following.

$$t_{med} = N \ln 2 / \lambda \quad (1)$$

Considering an overlay with one million nodes, distributed in 5000 ASes with 3 agents each group, the median session time of 15 minutes for agents, the churn rate for 15,000 agents is 11 per second by Formula 1. The total bandwidth to deliver the information within a group by router-based multicast protocol is 44Bps for downstream. As for the dissemination trees, the bandwidth is 220Bps for 5 fanouts per agents and middle leaders. By assuming average 20 nodes a team, average query rate of 0.5 per node/second, piggybacking three agents information each time, and 20 bytes per message, Table 1 shows the break down of bandwidth of every role and operation. As Table 1 shows, the extra overhead for agents and leaders is trivial. Even the load of agents or leaders are highly skewed, the overhead is not a burden for most of peers with DSL or cable modem connections.

## 4 Simulation and Experimental Results

In this section, we compare our design to Chord by simulation with real Internet AS-level topology and latency.

The Internet latency datum used in the simulation is from the King method [11], which has measured the network latencies of more than 1700 DNS servers. The AS-level topology graph is from the CIDR Report [6]. By mapping each DNS server to its AS, a 1701 nodes (ASes) graph with their network latency and topology has been formed. The average round trip delay between nodes pairs is 168ms, and the average AS path length is 2.97 hops. The p2psim [12] has been used to simulate the Chord protocol with Proximity Neighbor Selection (PNS). All experiments were performed in a Dell Dimension PC, which has one 2.8GHz Pentium IV processor and 1.2GB RAM, running Linux.

In order to make the experiments close to the deployment environment, the density and distribution of nodes of real P2P overlay are needed. The previous research [4] has showed that the average nodes for KaZaA and Gnutella were about 200 and 60 per AS, respectively. Also, all nodes distributed in about 5000 ASes. The host density, connectivity and traffic volume of P2P systems are highly skewed and exhibited heavy tails, which can be approximated by Zipf's distribution. As a result, our experiments are modeled to have 10,000 nodes over 100 ASes under the Zipf distribution. The 100 ASes is carefully chosen from the 1701 ones to keep almost equal latency and AS path length with the original one. For the Zipf distribution, the largest AS has 1,000 nodes and the smallest one has 36 ones. Also, the uniform distribution has been simulated for comparison.

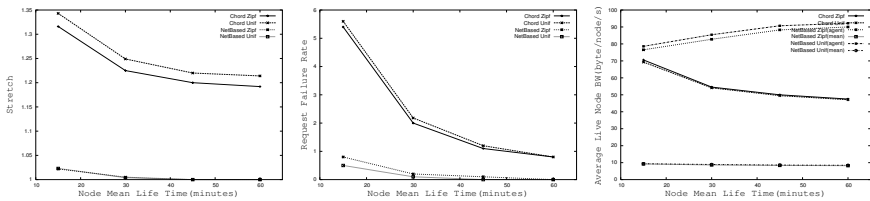
We use metrics of lookup latency stretch and failure rate for overlay performance, and bandwidth per node and link stress for overhead. In the simulation, each node alternately crashed and re-joined the network; the interval between successive events for

each is exponentially distributed with a mean time from 15 minutes to 1 hour. Every node issues lookups for random keys at intervals exponentially distributed with a mean of 10 seconds. The p2psim is configured with successors number of 16, base of 16 and refreshing intervals of 1 minute for both successors and fingers. Actually, this is a normal and modest configuration without favoring either request successful rate or consumed bandwidth, according to the research of Li et al. [12]. Our overlay is configured to have 3 agents for a group. The maximum nodes number of a team is 19, each leader is directly connected to an agent, and each message can piggyback up to 3 agents' IP addresses.

### 4.1 Experimental Results

The picture on the left of Figure 2 shows the result of the comparison of lookup latency stretch of the two overlays. The stretch is the ratio between the latency of the overlay routing to the ideal one. All protocols time out individual messages after an interval of three times the round-trip time, and the message is considered failure. The stretches include both successful and failure ones. As the figure shows, compared with the Chord protocol, our overlay is much more efficient even highly dynamic environment, the latency stretch is nearly the ideal one. Actually, the stretch under uniform distribution is a little better than skewed one for slightly higher successful rate. However, the skewed node distribution favors the Chord protocol with PNS supporting, since an AS with lots of nodes can facilitate a message to quickly approach the destination without traipse. Although the Chord protocol with PNS gets the benefit from skewed nodes distribution, its stretch increases quickly with the extent of system churn.

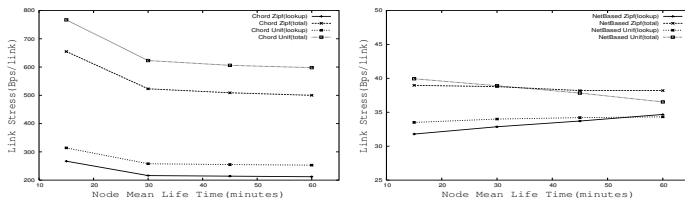
The picture in the middle of Figure 2 shows the request failure rate of different protocols. The system churn significantly impacts all P2P overlays. As the figure shows, the request failure rates of all protocols increase quickly with churn. Although the Chord with PNS can make a little benefit from skewed node distribution, its failure rate rapidly raises to more than 5% under 15 minutes mean node life time. Opposite to the Chord protocol, the failure rate of our protocol under uniform distribution is evidently better than skewed one. This is because that skewed distribution may exacerbate the churn in our overlay, since it will be harder to find more stable nodes for agents and leaders in some very small groups. Although the failure rate of our protocol is also increased fast, it will be no more than 1%. Also, our design is much better than Chord under all



**Fig. 2.** Comparison of stretch, request failure rate and bandwidth of different protocols and roles, under different system churn rates and node distributions

configurations. We argue that to involve unstable nodes into routing may not be a good choice, though the DHT can tolerate them.

The picture on the right of Figure 2 shows the bandwidth consumed by different protocols and roles. Since our protocol is asymmetric one, the bandwidth of both agents and the whole overlay are illustrated. Although the bandwidth of agents of our overlay is significantly larger than Chord, it is not a burden for most peers with high speed network connections as analyzed early, and they are no more than 3% of the overlay. For the mean bandwidth of the overlay, our protocol is highly efficient compared with Chord, and it is not sensitive to system churn.



**Fig. 3.** Link stresses put by different protocols to the Internet back bone, under different system churn rates and node distributions. The left one is for Chord protocol, the right one is for network based one.

While the bandwidth reflects the overhead of each peer, it can not indicate the traffic of the whole overlay. By mapping each node to its exact AS number and finding out the paths between them, we evaluate the traffic impact to the Internet back bones. By recording each link used by every message, the link stress has been computed. Figure 3 shows the link stress of different protocols. Both the total overlay link stress and the lookup (useful) one are presented. As the figure for Chord illustrates, system churn will significantly increase the P2P traffic in the Internet. Compared with uniform node distribution, skewed one generates less traffic due to more communication within one AS. However, Chord protocol is far from efficient. Not only is the average traffic value 12 to 20 times larger than ours, but also the useful (lookup) one is just about 40% of all. In another words, the useless (maintenance) overhead consumes most resource of the whole overlay. The link stress of our protocol is much less than the Chord, since most of maintenance work is accomplished with the same network. Moreover, the useful (lookup) part is more than 80%.

## 5 Related Work

In order to improve routing efficiency, physical network information is exploited in many DHT designs. Current works can be classified into three main categories: proximity routing, topology-based node ID assignment and proximity neighbor selection [13].

Proximity routing is employed in Chord [9] and their improvements. While physical network information is not taken into account when building system overlay, heuristic algorithms are used to choose many hops with small latency instead of large latency

ones. Topology-based node ID assignment is employed in CAN. When a new node joins the overlay, it joins a node that is close to it in IP distance. Proximity neighbor selection is employed in Pastry [14]. Routing table entries are selected according to the proximity metric among all peers that satisfy the constraints of logical overlay.

In Brocade [15], a secondary overlay network of supernodes based on AS-level topology is used to improve routing performance. Nodes in the default network establish direct connection to a supernode nearby. At the same time, supernodes collect the information of connected nodes and advertise their information in the second overlay. Although their benefits are limited by logical overlay restrictions, those systems produce significant improvements compared to original designs.

Some constant time lookup schemes have been proposed for P2P environment. Gupta et al. [16] proposed to maintain accurate routing table with complete membership in every node through quickly information dissemination. Although their design could reach the optimal stretch, the overhead to maintain the routing tables in each node was real expensive. Kelips [17] and HiScamp [18] were proposed to use gossip-style protocol and keep more states within each node to solve a key within  $O(1)$  hops. However, their convergence time for an event, such as node's joining and leaving, was too long.

## 6 Conclusions

Aware of the difficulties faced by current structure P2P designs, from a different angle, this paper proposes a new approach to build P2P overlay. Our approach focuses on the physical network, and builds the overlay closely matching it. Thus physical network characteristics, such as the power law of the Internet, network locality and asymmetric network connections, are naturally and fully exploited. The whole system not only keeps routing highly efficient, but also adapts extremely system churn by low maintenance overhead.

## References

1. S. Saroiu, P. K. Gummadi, and S. D. Gribble, "A Measurement Study of Peer-to-Peer File Sharing Systems," in *Proceedings of Multimedia Computing and Networking 2002 (MMCN '02)*, (San Jose, CA), January 2002.
2. F. Dabek, J. Li, E. Sit, J. Robertson, M. F. Kaashoek, and R. Morris, "Designing a DHT for low latency and high throughput," in *USENIX First Symposium on Networked Systems Design and Implementation (NSDI'04)*, Mar. 2004.
3. S. Rhea, D. Geels, T. Roscoe, and J. Kubiatowicz, "Handling Churn in a DHT," in *Proceedings of the USENIX Annual Technical Conference*, 2004.
4. S. Sen and J. Wang, "Analyzing Peer-to-Peer Traffic Across Large Networks," in *In Proc. ACM SIGCOMM Internet Measurement Workshop, Marseille, France, Nov. 2002.*, 2002.
5. Routeviews.org, "Route Views Archive." <http://www.routeviews.org>.
6. CIDR-Report, "The CIDR Report." <http://www.cidr-report.org>.
7. S. Ratnasamy, M. Handley, R. Karp, and S. Shenker, "Topologically-aware overlay construction and server selection," in *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Society (INFOCOM-02)*, 6 2002.
8. R. Cox, F. Dabek, F. Kaashoek, J. Li, and R. Morris, "Vivaldi: A Decentralized Network Coordinate System," in *Proceedings of the 2004 conference on applications, technologies, architectures, and protocols for computer communications (SIGCOMM)*, 2004.

9. I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in *Proceedings of the 2001 conference on applications, technologies, architectures, and protocols for computer communications (SIGCOMM)*, (San Diego, CA), pp. 149–160, 2001.
10. L. M., A. S., and F. A., "A Efficient Algorithms to Implement Unreliable Failure Detectors in Parially Synchronous Systems," in *Proceedings of the 13th Symposium on Distributed Computing (DISC'99)*, (Bratislava, Slovakia), 1999.
11. K. P. Gummadi, S. Saroiu, and S. D. Gribble, "King: Estimating Latency between Arbitrary Internet End Hosts," in *Proceedings of the SIGCOMM Internet Measurement Workshop (IMW 2002)*, (Marseille, France), November 2002.
12. J. Li, J. Stribling, R. Morris, M. Kaashoek, and T. Gil, "A performance vs. cost framework for evaluating DHT design tradeoffs under churn," in *Proceedings of 24th IEEE INFOCOM*, March 2005.
13. M. Castro, P. Druschel, Y. C. Hu, and A. Rowstron, "Exploiting network proximity in Peer-to-Peer overlay networks," in *International Workshop on Future Directions in Distributed Computing (FuDiCo)*, 2002.
14. A. Rowstron and P. Druschel, "Pastry: Scalable, decentraized object location and routing for large-scale peer-to-peer systems," in *Proceedings of the 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, (Heidelberg, Germany), Nov. 2001.
15. B. Zhao, Y. Duan, L. Huang, A. Joseph, and J. Kubiawicz, "Brocade: Landmark routing on overlay networks," in *Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS '02)*, (Cambridge, MA), 2002.
16. A. Gupta, B. Liskov, and R. Rodrigues, "Efficient Routing for Peer-to-Peer Overlays," in *USENIX First Symposium on Networked Systems Design and Implementation(NSDI'04)*, Mar. 2004.
17. I. Gupta, K. Birman, P. Linga, A. Demers, and R. van Renesse, "Kelips: Building an Efficient and Stable P2P DHT Through Increased Memory and Background Overhead," in *Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS'03)*, (Berkeley, CA), 2003.
18. A. J. Ganesh, A.-M. Kermarrec, and L. Massoulie, "HiScamp: self-organizing hierarchical membership protocol," in *Proceedings of the 10th European ACM SIGOPS workshop*, Sept. 2002.

# Dynamic Object Assignment in Object-Based Storage Devices

Lingjun Qin and Dan Feng

Key Laboratory of Data Storage Systems, Ministry of Education of China,  
School of Computer, Huazhong University of Science and Technology, Wuhan, China  
dfeng@hust.edu.cn, qinlingjun@yahoo.com.cn

**Abstract.** Object-based Storage Devices (OSDs) are the building block of Object-based Storage Systems. Object assignment in OSDs can largely affect the performance of OSDs. A real-time object assignment algorithm is proposed in the paper. The algorithm aims at minimizing the variance of Mean Response Time (MRT) across disks. To address the online problem with *a priori* unknown workload parameters, the algorithm employs an adaptive mechanism to estimate the parameters of workloads. The simulation results show that the algorithm can effectively balance the MRT in the disk I/O sub-systems.

## 1 Introduction

Object-Based Storage (OBS) is the new trends in network storage field [1]. Combining the benefits of the NAS and SAN architecture, OBS is allowing storage systems to reach petaBytes-scale levels. As the building block of OBS, Object-based Storage Devices (OSDs) play an important role in OBS and have great effects on the overall performance of the storage systems. Many efforts have been made to improve the performance of OSDs. RAID is one of the performance enhancing techniques that have been studied widely [2]. Through data striping, one object can be distributed among multiple disks to achieve parallel data transfer. Another important technology is object assignment. In order to remove the system bottleneck, objects should also be uniformly allocated among all the disks, thus balancing the load in a disk sub-system.

In on-line storage applications built on OBS systems, real-time object assignment is an essential issue which affects the overall performance. In such applications, data objects usually need to be stored immediately they are produced. For example, in high-energy physics, a particle accelerator can produce 100MB raw data per second. Moreover, these large amount of data should be stored in OSDs as quickly as possible for scientific research. This requires parallel I/O sub-systems to support on-line storage, involving providing a real-time algorithm for assigning objects to multiple disks.

Many data assignment algorithms have been proposed [3,4,5,6,7]. However, these algorithms employ off-line methods. Although off-line algorithms can produce optimized results, they are not suitable for real-time environment. Lin-Wen Lee et al. [6] have introduced an on-line algorithm for file assignment based on Mean Response Time (MRT), and assumed that the workload characteristics are known in advance and the service time of files is fixed. However, in some situation, it is unreasonable to



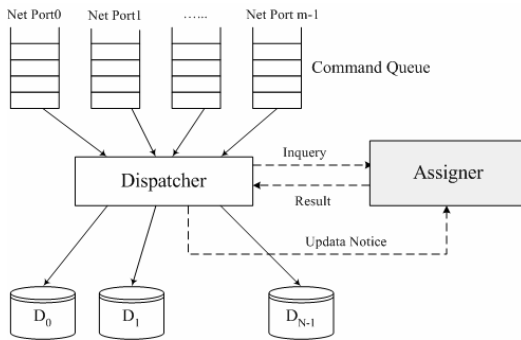
suppose that the characteristics of workload are known *a priori*. To tackle this problem, Scheuermann P. et al. [7] present a method to keep tracking of the change of workload and balance the heat (access rate) of disks using temperature (ratio between block heat and block size) as the criterion for selecting blocks in disks to be reallocated. Balancing heat of disks can reduce system response time, but only consider the access rate as the metric of response time is insufficient. To evaluate the response time of disks, we also need to consider the service time of objects, as well as the access rate of objects.

In this paper, we propose a dynamic algorithm for object assignment adopting MRT as cost function in real-time environment. Different from other algorithms, our algorithm can learn the parameters of workload and adaptively update information according to object access history.

The rest of this paper is organized as follow: In section 2 we introduce the object assignment in OSDs. Section 3 describes the dynamic assignment algorithm. Section 4 gives the simulation results and analyses. At last, we conclude the paper.

## 2 Object Assignment in OSD

OSD is a real-time system which exports an object-based interface. It contains processor, RAM, multiple Gigabit Ethernet channels and multiple high-speed disk drives, allowing it to process OSD commands (e.g., CREATE\_OBJECT, READ\_OBJECT and WRITE\_OBJECT [8]) and perform much more sophisticated object management.



**Fig. 1.** Object Assignment in OSD.  $m$  is the total number of network ports, and  $N$  is the total number of disks in OSD

The process of OSD commands is shown in Fig. 1. Each network port has an I/O command queue, and all the commands are scheduled by the Dispatcher. When a CREATE\_OBJECT command is processed, the Dispatcher should assign one of the disks for the incoming object. The Assigner is the decision maker for object assignment. It returns the most optimized results in response to the inquiry of the Dispatcher. The Dispatcher also sends update notice to the Assigner after processing an I/O command, so the Assigner can maintain up-to-date information.

The goal of assignment is to minimize the MRT of disks and reduce the variance of MRT between disks. The Assigner uses the dynamic greedy assignment policy to choose a disk with the minimal MRT. In this way, we can do real-time load balancing among all the disks.

Usually, an M/G/1 queue is used to model a single disk. Suppose that  $D_k$  ( $0 \leq k \leq N-1$ ) denotes the  $k^{\text{th}}$  disk in OSD, and the bandwidth of the  $D_k$  is  $B_k$ . Let  $OS_k$  be the set of indices corresponding to the objects which are stored in  $D_k$ , and  $O_{k,i}$  be the  $i^{\text{th}}$  object in  $D_k$ . According to the model of M/G/1, the MRT of  $D_k$  can be given as following three formulas [6]:

$$MRT_k = E(T_{w,k}) + E(T_{s,k}) = \frac{\lambda_k E(T_{s,k}^2)}{2[1 - \lambda_k E(T_{s,k})]} + E(T_{s,k}), \tag{1}$$

$$E(T_{s,k}) = \sum_{i \in OS_k} \lambda_{k,i} \frac{S_{k,i}}{B_k} / \sum_{i \in OS_k} \lambda_{k,i}, \tag{2}$$

$$E(T_{s,k}^2) = \sum_{i \in OS_k} \lambda_{k,i} \left( \frac{S_{k,i}}{B_k} \right)^2 / \sum_{i \in OS_k} \lambda_{k,i}. \tag{3}$$

$T_{r,k}$ ,  $T_{w,k}$  and  $T_{s,k}$  denote the response time, wait time and service time of  $D_k$  respectively.  $\lambda_k$  is the total object access rate in  $D_k$ , and  $\lambda_k = \sum_{i \in OS_k} \lambda_{k,i}$ , where  $\lambda_{k,i}$  is the access rate for  $O_{k,i}$ .  $S_{k,i}$  denotes the mean request size for  $O_{k,i}$ . From the formula (1)~(3), we get:

$$MRT_k = \frac{1}{B_k} \left[ \frac{P_k}{2(B_k - Q_k)} + \frac{Q_k}{\lambda_k} \right]. \tag{4}$$

Here  $P_k = \sum_{i \in SO_k} \lambda_{k,i} S_{k,i}^2$  and  $Q_k = \sum_{i \in SO_k} \lambda_{k,i} S_{k,i}$ . Formula (4) is the cost function for object assignment. OSD maintains the real-time value  $MRT_k$  ( $0 \leq k \leq N-1$ ) for each disk. When disk allocation is requested, OSD selects the disk with the minimal MRT.

Note that the value of  $MRT_k$  is computed by  $\lambda_{k,i}$  and  $S_{k,i}$ . Although we know nothing about the characteristics of workload, we use an adaptive method to learn and dynamically update the value of  $\lambda_{k,i}$  and  $S_{k,i}$ .

### 3 Dynamic Object Assignment Algorithm

The load characteristics of objects need to be tracked dynamically since they change with time. To do this, OSD records the history access information of  $O_{k,i}$  within a moving window which has the length  $WL_{k,i}$ .

We treat the history information as attributes of object. Here we define a new attribute page following the T10 OSD protocol to save this information [8]. The page, named Access Attribute Page, is shown in Fig. 2.

In Fig. 2, the attributes with subscript “ $k,i$ ” denotes the attributes belonging to the  $i^{\text{th}}$  object in  $D_k$ . Array  $A_{k,i}[j]$  and  $B_{k,i}[j]$  ( $0 \leq j \leq M_{k,i}-1$ ) store the access time and request size for the same request. Here  $M_{k,i}$  is the maximal length of array.  $M_{k,i}$  can be determined by  $M_{k,i} \leq WL_{k,i} \cdot \text{MAX}(\lambda_{k,i})$ , where  $\text{MAX}(\lambda_{k,i})$  is the maximal possible access rate of  $O_{k,i}$ . The pointer  $p_{k,i}$  is the index of array  $A_{k,i}$  and  $B_{k,i}$ , and points to the latest request information within  $WL_{k,i}$ . The counter  $c_{k,i}$  is the total access number of the object in  $WL_{k,i}$ .

We also need to store the up-to-date value of  $Q_k$ ,  $P_k$  and  $MRT_k$ . Because they are only related to  $D_k$ , we save them in attribute page of Root object in  $D_k$ .

Attribute Number	Attribute
0	Access Attribute Page ID
1	$\lambda_{k,i}$
2	$S_{k,i}$
3	$WL_{k,i}$
4	$p_{k,i}$
5	$c_{k,i}$
6	$M_{k,i}$
7	$A_{k,i}[0], B_{k,i}[0]$
...	...
$M_{k,i}+6$	$A_{k,i}[M_{k,i}-1], B_{k,i}[M_{k,i}-1]$

Fig. 2. Access Attribute Page for  $O_{k,i}$

Using the attributes of objects, the Assigner automatically learns load characteristics and adaptively assigns objects. When inquired by the Dispatcher, the Assigner should return a disk ID with the minimal MRT. While I/O command except for CREATE\_OBJECT is processed, the Assigner will be informed to update values of  $\lambda_{k,i}$  and  $S_{k,i}$  and recompute  $MRT_k$ . The estimated value of  $\lambda_{k,i}$  can be obtained from the access frequency within  $WL_{k,i}$ . Similarly,  $S_{k,i}$  can be get from the average request size within  $WL_{k,i}$ .

The message passing between the Dispatcher and the Assigner contains following items:

- *Type*: The type of the message;
- $O_i$ : The object to be accessed, and  $i$  is the index of the object;
- $k$ : The index of disk which  $O_i$  is located in;
- $O$ : The object to be created;
- $m$ : The index of disk which  $O$  will be assigned to;
- $CT$ : Current time;
- $RS$ : Request size for  $O_i$  in  $D_k$

The pseudocode for the dynamic assignment algorithm used in the Assigner is depicted as follows:

```

for(;;){
  Receive message from the Dispatcher;
  if(Type ≠ CREATE_OBJECT_NOTICE){
    Get  $Q_k$  and  $P_k$  from Root object attribute page;
    Get Access Attribute Page associated with  $O_i$  in  $D_k$ , as is shown in Fig. 2;
  }
  switch(Type){
    case CREATE_OBJECT_NOTICE:
      For  $0 \leq j \leq N-1$ , get  $MRT_j$  from Root object attribute page in  $D_j$ ;
       $MRT_m = \text{MIN}\{MRT_j \mid 0 \leq j \leq N-1 \text{ and } Q_j/B_j < 1\}$ ;
      if(Can not find  $MRT_m$ )
        exit(); /* OSD is overloaded*/
      Assign  $O$  to  $D_m$  (Suppose the index of  $O$  is  $l$  after  $O$  is created);
      Create Access Attribute Page associated with  $O_l$  in  $D_m$ ;
       $\lambda_{m,l} = S_{m,l} = p_{m,l} = c_{m,l} = 0$ ;
      Initialize  $M_{m,l}$  and  $WL_{m,l}$  according to the type of  $O$ ;
      Initialize  $A_{m,l}[j]$  and  $B_{m,l}[j]$  ( $0 \leq j \leq M_{m,l}-1$ );
      break;
    case REMOVE_OBJECT_NOTICE:
       $Q_k = Q_k - \lambda_{k,i} S_{k,i}$ ;  $P_k = P_k - \lambda_{k,i} S_{k,i}^2$ ;
      Update  $MRT_k$  using formula (4);
      break;
    case READ_OBJECT_NOTICE:
    case WRITE_OBJECT_NOTICE:
       $X = \Phi$ ;  $c = c_{k,i}$ ;
      if( $p_{k,i} + 1 < c_{k,i}$ )
        init_index =  $p_{k,i} + M_{k,i} - c_{k,i} + 1$ ;
      else
        init_index =  $p_{k,i} - c_{k,i} + 1$ ;
       $p_{k,i} = p_{k,i} + 1$ ;  $A_{k,i}[p_{k,i}] = CT$ ;  $B_{k,i}[p_{k,i}] = RS$ ;  $c_{k,i} = c_{k,i} + 1$ ;
      index = init_index;
      while(index ≠ init_index){
        if( $A_{k,i}[\text{index}] < CT - WL_{k,i}$ ){
           $c_{k,i} = c_{k,i} - 1$ ;
          index = (index + 1) %  $M_{k,i}$ ;
           $X = X \cup \{\text{index}\}$ ;
        }
      }
       $\lambda'_{k,i} = c_{k,i} / WL_{k,i}$ ;  $S'_{k,i} = (S_{k,i} \square c - \sum_{j \in X} B_{k,i}[j] + RS) / c_{k,i}$ ;
       $Q_k = Q_k - \lambda_{k,i} S_{k,i} + \lambda'_{k,i} S'_{k,i}$ ;  $P_k = P_k - \lambda_{k,i} S_{k,i}^2 + \lambda'_{k,i} S_{k,i}'^2$ ;
       $\lambda_{k,i} = \lambda'_{k,i}$ ;  $S_{k,i} = S'_{k,i}$ ;
      Update  $MRT_k$  using formula (4);
  }
}

```

It should be pointed out that different types of objects need different  $WL_{k,i}$ . For example, a multimedia object should have a short  $WL_{k,i}$  since users access pattern changes frequently, whereas a history archive object needs a long  $WL_{k,i}$  due to its low access frequency.

Note that the utilization of  $D_k$  is  $Q_k/B_k$ . We avoid assigning objects to an overloaded disk when its utilization is close to 1 and assign object to the disk which satisfy the condition  $Q_k/B_k < 1$ .

### 4 Simulation Results and Analyses

The simulation is based on the synthetic workload. In our tests,  $NO=5000$  objects are partitioned to  $N=4$  disks. The objects are numbered from 0 to 4999, and created in OSD with the ascending order. For the sake of simplification, objects will not be removed from OSD after they are assigned to disks. In order to generate unbalanced MRT across disks, the access rate and request size of objects are distributed according to exponent distribution. To make sure that the utilization of each  $D_k$  ( $0 \leq k \leq N-1$ ) is less than 1, the mean values of the access rate and request size of objects in  $D_k$ , that is  $\bar{\lambda}_k$  and  $\bar{S}_k$ , will satisfy the following condition:  $\bar{\lambda}_k \bar{S}_k < \frac{N}{NO} B_k$ . In our experiments, we set  $\bar{\lambda}_k = 2$  (accesses/minute),  $\bar{S}_k = 20$  (ms). We also assume that all the disks have the same bandwidth, that is  $B_k=1$ .

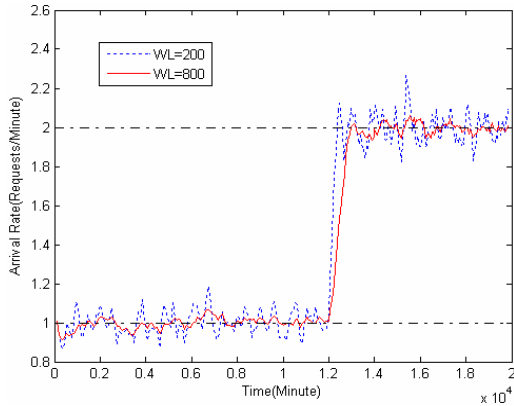


Fig. 3. Workload characteristics tracking

In order to evaluate the effect of workload tracking, we produce an access sequence. For each object, we generate a Poisson arrival sequence according to its arrival rate. The item in the sequence indicates the time the object will be accessed. We also associate each item with a service time according to the mean service time of the object. Then we synthesize all the sequences into one big sequence in ascending time order and input it into our simulation program. The program reads the sequence and gets the object access time and service time, and dynamically estimates the mean

arrival rate and mean service time for each object. Fig. 3 plots the arrival rate estimates for one object. The object has the changing workload with Poisson request arrival rate. At first the mean arrival rate  $\bar{\lambda}$  is 1, and then changes to  $\bar{\lambda}=2$ . We can see that the estimated value fluctuates around its actual value, which means that our algorithm can successfully learn the workload. In order to show the influence of the length of the moving window ( $WL$ ), we set two window length, that is  $WL=200$ (minutes) and  $WL=800$ . Fig. 3 tells us that the longer the length of window is, the more precise the estimated value can get. But the long window also leads to low sensitivity to the change of workload. In Fig. 3, the curve of  $WL=200$  keeps up with the changing workload more quickly (but with larger fluctuation) than the curve of  $WL=800$ . So it is important to choose an appropriate length of moving window.

We compare the dynamic assignment algorithm with a random assignment algorithm. The MRT of each disk and the covariance of MRT of all the disks are plotted in object creating order as shown in Fig. 4 and 5. The results show that, under dynamic algorithm, the covariance of MRT among disks tends to be smaller, while the covariance under random algorithm rapidly increases with more and more object being assigned to disks. It is worth to notice that there is some fluctuation at the beginning of curve. This is because the accumulated MRT of each disk is small at first, and any coming object makes the MRT change drastically.

We also study the efficiency of the dynamic object assignment when the workload characteristics changes. To do this, we randomly change the access rate and request size of all the assigned objects after the 5000<sup>th</sup> object is assigned. Due to the change of workload, the MRT of each disk is markedly different from others. Then we generate a new workload with another 4000 new objects and continue to assign these objects to disks. In this way we evaluate the adaptability of our algorithm, and the results are drawn in Fig. 6.

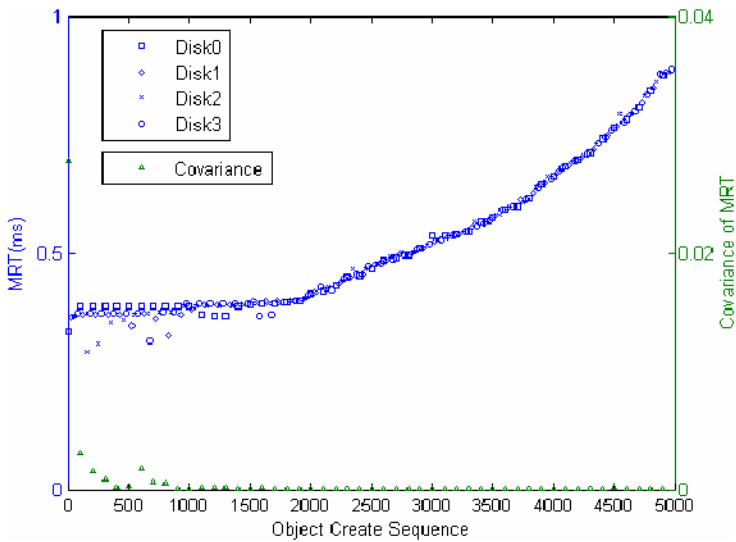
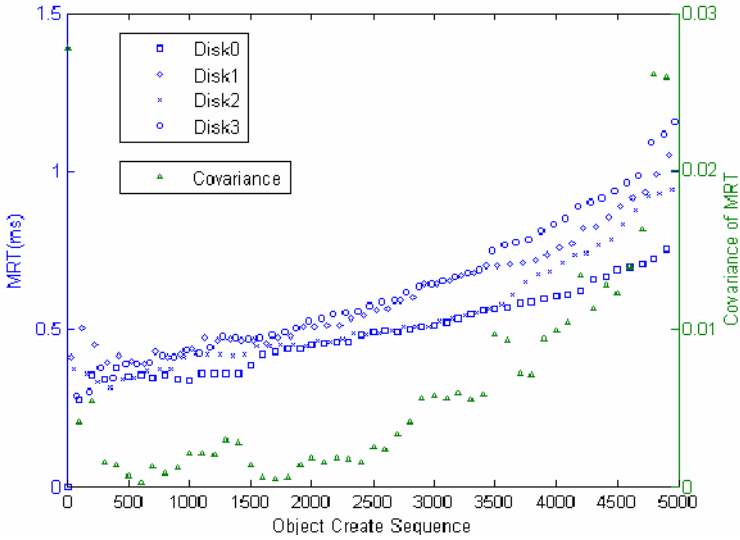
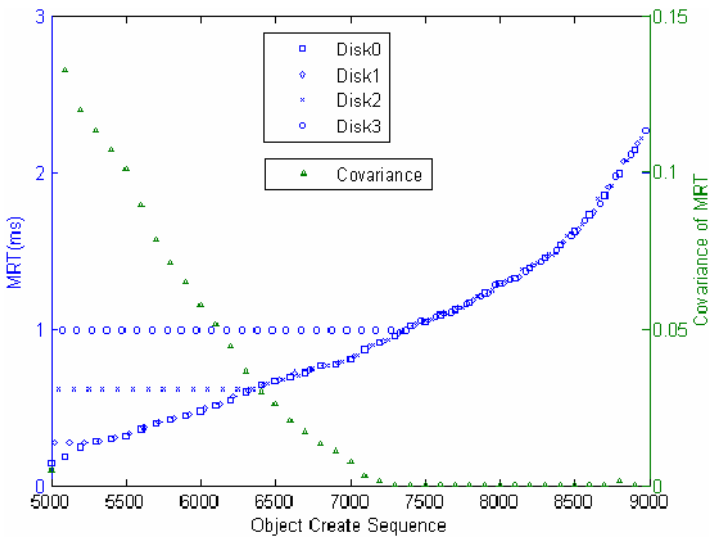


Fig. 4. Effect of dynamic algorithm



**Fig. 5.** Effect of random algorithm



**Fig. 6.** Effect of dynamic algorithm after workload changes

From Fig. 6, we can see that the algorithm quickly learns the new workload parameters and has the ability to balance the MRT among disks. With more objects assigned, the covariance of MRT drops to a low value, despite the covariance was large before.

## 5 Conclusions

In this paper, we have presented a dynamic object assignment algorithm that aims at minimizing the covariance of Mean Response Time (MRT). The algorithm can learn the parameter of the workload under the real-time environment with the changing workload. The simulation results show that the algorithm can effectively balance the MRT across disks.

## Acknowledgements

This work was supported by “973” project of China No.2004CB318201, National Science Foundation of China No.60273074 and Huo Yingdong Education Foundation No.91068.

## References

1. Mesnier M., Ganger G.R., Riedel, E.: Object-based Storage. *IEEE Communications Magazine*, Vol 41, No. 8. (2003)84-91
2. Sahai, A.K.: Performance aspects of RAID architectures. *Performance, Computing, and Communications Conference*, Phoenix, USA. (1997) 321-327
3. Pattipati K.R., and Wolf, J.L.: A file assignment problem model for extended local area network environments. *10th International Conference on Distributed Computing Systems*, Paris, France. (1990) 554-561
4. Xiangwu Meng, and Hu Cheng: Solving file allocation problem based on genetic algorithms. *Journal of Software*, Vol 8, No. 2. (1997) 122-127
5. Deng-Jyi Chen, Ruey-Shun Chen, Hol W.C., and Ku K.L.: A heuristic algorithm for the reliability-oriented file assignment in a distributed computing system. *International Conference on Parallel and Distributed Systems*, Hsinchu, Taiwan. (1994) 454-459
6. Lin-Wen Lee, Scheuermann P., Vingralek R.: File assignment in parallel I/O systems with minimal variance of service time, *IEEE Transactions on Computers*, Vol 49, No.2. (2000) 127-140
7. Scheuermann P., Weikum G., Zabback P.: Data partitioning and load balancing in parallel disk systems, *The VLDB Journal - The International Journal on Very Large Data Bases*, Vol. 7, No. 1. (1998) 48-66
8. Draft OSD Standard, T10 Committee, Storage Networking Industry Association (SNIA), <ftp://ftp.t10.org/t10/drafts/osd/osd2r00.pdf>



# Dynamic Resource Discovery for Sensor Networks<sup>\*</sup>

Sameer Tilak<sup>1</sup>, Kenneth Chiu<sup>1</sup>, Nael B. Abu-Ghazaleh<sup>1</sup>, and Tony Fountain<sup>2</sup>

<sup>1</sup> Computer Science Department, SUNY Binghamton

<sup>2</sup> San Diego Supercomputer Center, University of California at San Diego

**Abstract.** As sensor networks mature the current generation of sensor networks that are application-specific and exposed only to a limited set of users will give way to heterogeneous sensor networks that are used dynamically by users that need them. The available sensors are likely to be dynamic (e.g., due to mobility) and heterogeneous in terms of their capabilities and software elements. They may provide different types of services and allow different configurability and access. A critical component in realizing such a vision is *dynamic resource discovery*. In this paper, we develop a resource discovery protocol for sensor networks, outline some of the challenges involved, and explore solutions to some of the most important ones. Specifically, we first discuss the problem of what resources to track and at what granularity: in addition to the individual sensor capabilities, some resources and services are associated with sensor networks as a whole, or with regions within the network. We also consider the design of the resource discovery protocol, and the inherent tradeoff between interoperability and energy efficiency.

## 1 Introduction

The success and increasing deployment of Wireless Sensor Networks (WSNs) is driving towards a new generation that envisions commodity sensor networks providing standard services that can be composed by clients into a wide range of applications. These sensor networks then form a critical, yet generic interface between the physical and digital worlds, converting physical qualities into measurements that can be then harnessed for a wide-ranging spectrum of applications. The impact of such a development would be profound: no longer would sensor networks be specialized networks serving a limited set of users, but rather a basic infrastructure supporting and encouraging unanticipated functions and modalities of operation. Increasingly advanced and accessible applications will be enabled as sensor networks are shared dynamically and ubiquitously, allowing clients to unite disparate components on demand into *virtual sensor networks*. Some clients may even compose services spanning multiple sensor networks into a single end-to-end service for use by applications.

---

<sup>\*</sup> This work was partially supported by NSF Awards SCI-0330568, CNS-0454298 and DBI-0446298, and Air Force grants FA8750-04-1-0211 and FA8750-05-1-0130.

Despite the recent emergence of sensor networks as a field of study, already many sensor hardware platforms with a wide range of capabilities have emerged. In addition, there are large differences in the software elements (operating systems, networking protocols, data base systems, etc.) used for these platforms. We believe that this heterogeneity is inevitable, and, furthermore, we believe that an attempt to impose uniformity on this diversity would stifle experimentation and innovation. But without some commonality, interoperability is not possible. We thus believe in the adoption of minimal, extensible standards that can promote interoperability by supporting the discovery of formats and protocols.

A necessary precursor to interoperability is the ability to discover beneficial resources and the attributes of these resources required for interoperation. Resource discovery in sensor networks is challenging for a number of reasons. Since different sensor networks are deployed and managed by different organizations, they are heterogeneous in aspects such as protocols, architectures, security policies, and management policies. Also, the nature of resources (sensing and coverage characteristics, connectivity characteristics, available energy, computational and storage capabilities, protocols and software elements) and services (e.g., localization, synchronization, calibration) are significantly different from traditional distributed systems. Furthermore, the potential mobility of sensors and clients introduce unique challenges [17]. Finally, the embedded nature of sensors place a premium on energy efficient solutions. Thus, WSN resource discovery requires significantly different solutions from traditional naming and resource discovery in distributed systems [1, 10].

The resource discovery problem in sensor networks can be broken down into two components: (1) Determining what resources to track and at what granularity to track them. In the context of sensor networks this is a challenging and multifaceted problem; and (2) determining the resolution protocol that queries the sensor network for the resource information and transfers this information to queries in an energy efficient way. Note that the two components are similar to other distributed resolution systems (e.g., DNS' servers and resolvers [15]). However, sensor network operation introduces many characteristics that make this problem unique such as resource attributes that are associated with the network as a whole or regions within, as well as energy efficiency concerns.

## 2 Dynamic Resource Discovery in Sensor Networks

An essential component in systems where foreign and heterogeneous elements interoperate is the ability to discover relevant information regarding available resources. A new client generally has no knowledge of what sensor resources exist around it. Moreover, properties such as protocols and formats that are relevant to interoperation must also be obtained. We call this problem *Dynamic Resource Discovery (DRD)*. This section overviews the challenges in DRD for sensor networks. The discussion is organized into the two major aspects of DRD: (1) Determining tracked attributes; and (2) Resource Discovery.

**Determining Tracked Attribute Set:** The decision on what resources, services and other system attributes are tracked to support interoperability and service discovery is an important one. The candidate attribute set for tracking may be classified along multiple axes. First, attributes can be classified into those that provide information regarding interoperability (e.g., protocols used or formats for messages), and those that describe the metastate of the system (resources or services provided). Standardization plays a role in determining what interoperability information to track (e.g., if sensors are standardized to be completely homogeneous, no interoperability information needs to be tracked). It is important to discriminate here between resource discovery and data collection. While both operations require collecting information from the sensors, DRD collects meta-information and not data. Although the resource discovery and data collection may be combined for efficiency in certain cases, it is likely that separating them will lead to a more effective and modular solution.

A different axis for classifying tracked attributes resource granularity. We classify attributes as simple (associated with single sensors) and complex (representing multiple sensors). Clearly, individual sensor resources are of interest. These include sensing resources (such as available sensors, their tolerances, and their coverage) as well as computational, storage and communication resources. Additional properties exist for the network as a whole (e.g., what routing protocols are used). Moreover, resources may be aggregated: instead of tracking and reporting sensor resources in detail, they may be summarized (e.g., coverage in an area instead of individual sensor location). Thus, the system must be able to associate and track resources at these different granularities.

A related issue arises due to the data-centric nature of sensor networks: they are embedded within the environment they are monitoring, and are mostly of interest only in terms of what information they can provide about the environment. Resources may be associated with regions in terms of network organization (e.g., resources or services in a cluster) or in terms of the environment being monitored (e.g., resources or services available in a room). The resource discovery protocol should provide the flexibility of tracking resources in terms of application-level or infrastructure-level organization. The choice of what attributes to track and at what granularity to track them is left to application developers and not discussed further in this paper.

**Resource Discovery Protocol:** Once the tracked attributed set is determined, the role of the resource discovery protocol is twofold: (1) Track the values of the attributes at selected points within the network; and (2) Respond to client queries. It is necessary to minimize the communication required for implementing DRD due to energy efficiency considerations. Energy efficiency also dictates being able to aggressively power down sensors when they are not being used; powered down sensors cannot receive or respond to queries.

In terms of tracking the selected attributes, a choice exists between pushing the attributes proactively to selected points in the network or pulling them in response to received queries. In distributed systems, generally pull is preferred to push when requests are infrequent relative to the data change rate, while

push is preferred when requests are frequent. In sensor networks, two additional factors favor at least partial push of resource information: (1) because sensors must operate on a low duty cycle, pulling the data causes large delays if the data is locally maintained at sensors which are currently sleeping; and (2) for complex attributes, the attributes must be summarized across multiple sensors; this requires pushing the data to points in the network where they can be combined.

Depending on the chosen approach, query resolution is implemented. In this component of the resource discovery protocol, the client query is forwarded to attribute repositories (locations in the network where attributes are collected) that are relevant to it. In a brute force unstructured approach, the queries may be flooded within the network. More efficient unstructured solutions may use probabilistic algorithms such as gossiping. Alternatively, if the resource space is structured such that the location of attributes relevant to a query are known then more efficient query forwarding can be used (e.g., name resolvers in DNS [15]). Further, caching may be used to optimize operation in either approach [20].

Interoperability favors a standardized protocol like ASCII-based XML that does not presume specialized protocols with potentially foreign elements generating the queries. However, efficiency dictates custom protocols that are compact. In the next two sections, we investigate alternatives to balance these requirements and show that it is possible to optimize the size of the exchange messages without sacrificing interoperability.

### 3 Architecture

To explore the feasibility of our vision of future sensor networks, we have developed a simulation-based prototype for DRD. We use an architecture where sensors self-organize to form clusters. A cluster is a collection of sensors that are associated/represented by a single *Cluster Head* (CH). For DRD, this organization serves the following purposes: (1) The CH represents a logical point for maintaining complex attributes; (2) The CH receives DRD queries and is able to respond to them, freeing the remaining sensors to be powered down in periods of inactivity; and (3) Finally, if hierarchical naming is desired, clusters can in turn themselves be clustered into bigger entities to provide more efficient query forwarding for structured attribute sets. We note that the use of clusters is common in sensor networks [11], mainly to allow data aggregation/reduction (similar to our complex attributes) and resource arbitration resulting in more scalable and energy efficient solutions.

In our algorithm, similar to GAF [30], cluster membership is determined geographically. The sensor field is divided into zones such that all sensors within a zone are in range with each other. Cluster selection is then localized to a zone such that a sensor only considers CH advertisements occurring in its zone; only one CH is selected per zone. We note that this approach requires either pre-configuration of the sensors or the presence of a location discovery mechanism (GPS cards or a distributed localization algorithm [3]). In sensor networks, localization is of fundamental importance as the physical context of the reporting

sensors must be known in order to interpret the data. We therefore argue that our assumption that sensors know their physical co-ordinates is realistic. We emphasize that cluster formation is orthogonal to the proposed resource discovery protocols and other cluster formation approaches can be used. The overall operation consists of the following phases:

1. Cluster head election: Cluster election identifies cluster heads, which serve as control points, data sinks and meta-data repositories for their cluster members. In this phase, sensors send Resource Description Format (RDF) [27] advertisements indicating their remaining energy to neighbors. Upon receiving all neighbor messages, the node with the highest remaining energy is elected as CH. Other selection criteria are possible.
2. Meta-data exchange: In this phase, nodes send their meta-data to the CH as an RDF message, where it is inserted into an SQL database using a simple mapping from RDF properties to table columns. For static attributes, this phase is combined with CH election, piggybacking meta-data on CH election messages. For dynamic attributes collection may be required more frequently. For example, if a video sensor changes its angle, zoom, or resolution such meta-data needs to be registered with each change.

Up till now, we have focused on the discovery protocols (attribute tracking). However, meta-data is also needed to describe the data stream reported by a sensor. Interestingly, for some long running queries, the data reported by a sensor might change more dynamically. Consider a sensor that has temperature and audio transducers on board. A query may require the sensor to report its data when any of the temperature or audio measurements have interesting events. In such cases, the sensor needs to associate meta-data with its data stream so that the receiver can identify whether its a temperature, audio, or a combination of these streams.

3. Query generation: In this phase, new clients that need to interact with the network pose queries to it. The query API is either standardized or negotiated via the resource discovery protocol. In our study, the query message is an SQL query sent as ASCII.
4. Query forwarding: In this phase, queries from clients are forwarded towards attribute repositories that can satisfy them. Depending on the application, there may be room for optimized query forwarding (e.g., for attributes that are strictly hierarchical, or when queries can benefit from the results of previous ones). For unstructured resources, multicast or epidemic algorithms are most efficient.
5. Query processing: A CH upon receiving a query, processes it and sends back the appropriate reply.
6. Resource/service usage: Based on the reply from the previous stage,  $S_1$  (or a client) starts accessing the requested resources and services. For example, the new sensor might find its current physical coordinates using localization service provided by location anchors and upon negotiating data transfer protocol, it might start reporting its observations to the new CH.

## 4 Architecture Discussion

DRD involves several aspects, including resource description, resource registration, resource query, and message encoding. The presented architecture describes the overall interaction of these aspects, but is not tied to the particulars of how resources are described, how queries are formulated, or how information is encoded. These can all be done in a number of ways, each with different trade-offs. Resource discovery in sensor networks can draw from a number of efforts in distributed systems and web services. In this section, we provide further discussion of some of the important facets of DRD.

**Resource description:** A number of possibilities exist for resource description. We could of course simply use a completely ad hoc resource model. This has the advantage that we can customize it to be very compact, and tailored for DRD needs. The disadvantage, though, is that it would be incompatible with other resource descriptions. This means that it would not be able to leverage other associated software and standards. Since we could not see any significant advantage to creating our own resource descriptions, we used the RDF model developed by the W3C. We anticipate that as our research matures, we will actually develop an ontology of resources. This will allow us to apply description logic to resources. As we develop an ontology, we will naturally move from RDF to OWL. This is aided by the fact that OWL actually uses RDF. We are currently not using the standard RDF/XML syntax, but we anticipate that we will as our work matures. We have addressed size concerns by using binary XML, which we describe below.

**Query formulation:** When a client needs to query the CH for the available resources, it needs to formulate the query in some language. We are currently map the RDF description to a table, and thus formulate the query as an ASCII SQL string. This has the advantage of using a common, simple language for queries, but cannot handle more complex, structured resource queries. Query formulation for resource descriptions and ontologies is currently an active research and development area. A number of RDF and OWL query languages exist, such as OWL-QL [8], RDQL [18], and RQL [14]. Further investigation is necessary to determine which approaches will work best for sensor networks.

**Formats and encodings:** A number of formats can be used to encode the messages used in the various phases. Since communication requires large energy expenditures, a compact format has the advantage of conserving energy. For example, the resources of a node may be represented as: (41 1000 0 500 1000). This data being 5 integers, takes only 20 bytes, but to interpret these numbers correctly, the client must know that the first number is the ID, the second number is the precision, the third number is the minimum of the range, the fourth number is the maximum of the range, and the last number is the capacity. Such detail is error-prone, and the format may lack flexibility and interoperability. Changes to the format will be hard to detect, may cause strange failures or hard-to-find bugs.

Formats such as XML, along with standards such as SOAP [26] and WSDL [6], offer the advantage of having a large community of developers and researchers working on a common, well-known language. By using these, we can leverage the standards and available expertise. The disadvantage of these formats, however, is that they are verbose. The verbosity incurs large energy costs during communication. An XML document typically contains many identical strings referring to tags, namespace prefixes, and namespace URIs. Every XML element also includes a redundant end tag. The following XML document contains the same data as the custom message above, but requires 191 bytes.

```
<?xml version='1.0' encoding='ISO-8859-1' ?>
<index version='1.0'>
  <id>41</id>
  <precision>100</precision>
  <minrange>0</minrange>
  <maxrange>500</maxrange>
  <capacity>1000</capacity>
</index>
```

One alternative that reduces the energy costs, but retains many of the advantages, is to use alternative encodings of XML. XML is specified as a textual format for structured data. The specification implicitly suggests an abstract data model for tree structured data consisting of parent nodes and child nodes, with their associated attribute, content, and other information. This model does not inherently preclude an efficient encoding, and thus alternate serializations of that are much more concise than textual XML. These alternative serializations, are commonly known as “binary” XML, since they often utilize the full 8-bit range of the constituent bytes. Since binary XML is logically the same as textual XML, a single API can work for both. Also, all standards based on XML work with binary XML, since most XML standards reference the XML abstract model known as XML Infoset.

We have tested two versions of a binary XML. The first version eliminates the redundant end tag, and replaces ASCII numbers with their two’s complement representations. The second version achieves further compactness by separating much of the static meta-data about the message from the data of the message itself. This is achieved by using a separate XML Schema of the message, and combining the Schema with a compact representation of the data to reconstitute the complete XML document. This is similar to ideas in P BIO [7] and DF DL [5].

Interpretation of these compact messages requires an associated schema. This association can occur through a number of mechanisms. One technique is to simply include a schema URI in the beginning of the message. This overhead may be too high for short messages, however, since the URI is typically a relatively long string. Another technique is to assume that some previous information has been exchanged, and then associated with the particular communication channel corresponding to the compact message. For our prototype, we have simply hard-coded this association. We note that this use of the schema to interpret a compact representation of the message can also be applied to textual XML.

## 5 Experimental Study

In this section we first describe the design of our simulation framework and then demonstrate its utility using a simple prototype simulation study. We hope to extend this framework as we develop our VSN subsystem to complement implementation on real sensors. The evaluation testbed is completely based on open source software components; either already available in public domain or built in house. The integration of these typical software components ranging from an embedded database, network simulator, and XML parsers provides an accurate development and evaluation environment that can reduce the development barriers for a broad spectrum of applications. It can also assist researchers in evaluating their protocols. We now describe the individual components of the framework.

- *SQLite database*. SQLite is a self-contained, embeddable, zero-configuration SQL database engine [19]. We decided to choose SQLite because it implements most of SQL92 and requires zero-configuration – no setup or administration is needed. This property is useful because the large scale and autonomous nature we target, prohibits manual configuration and administration. It stores the complete database in a single disk file. This is important given the resource constraints – having multiple database files on disk and memory may not be possible for some embedded sensor filesystems. Most important to our purposes, it has a small code footprint.
- *Libxml2*. It is the XML C parser [24] and toolkit developed for the Gnome project. We are also exploring various other XML parsers such as TinyXml [22], XPP [29], Xerces [28] and would like to evaluate their performance and power characteristics.
- *Binary xml parser*. We used an experimental binary XML implementation developed in-house, which is discussed in Section 4.
- *ns-2*: It is an open source discrete event network simulator [16]. In our simulations we used a CSMA based MAC layer and our power analysis is based on the energy model pre-built in ns-2.

This framework does not imply that there exists heterogeneity in terms of software elements. Specifically, we do not expect individual software components (SQLite, libxml etc.) to be installed on all the sensors. However, we expect the system components to follow appropriate standards and expose standardized interfaces. For example, a mote might run an instance of TinyDB [21] database, whereas a PASTA node might run an instance of SQLite. However, since both these database technologies follow standards and expose SQL interface, they fit in within our system design notion. To summarize, we take a technology agnostic view and our system design is based on open standards and standardized interfaces. Above mentioned software components just provide a concrete basis to build our simulation framework, and not as an end in itself.

At some level in the system, commonality must exist. From this common level, clients would then bootstrap and configure themselves to work with the relevant sensors. For the lowest levels, we believe that RDF is the appropriate



**Table 1.** Energy consumption study

Protocol	Data size (bits)	Transmit Energy (J)	Receive Energy (J)
custom	160	0.000033	0.000038
ASCII xml	1448	0.000302	0.000342
binary xml	160	0.000033	0.000038

choice. Above that, however, we believe that the issue is an open question which we will address in future work. We also believe that connectivity to the grid and web services will be important. Such interoperability can greatly increase the effectiveness and utility of sensor networks by allowing them to be plugged into wide area cyberinfrastructures.

To demonstrate the utility of the proposed framework, we conducted prototype study to evaluate the energy efficiency of message encoding format alternatives. For modeling energy model realistically, we used a 802.15.4 style low power radio: the bandwidth is 250kbps, and the transmit, receive and idle power are 0.0522, 0.0591 and 0.00006 Watts respectively. The simulation area was set to  $350 \times 350 m^2$  with 50 sensor nodes. Each zone was  $70 \times 70 m^2$  (total 25 zones). Radio transmission range was set to 100  $m$  to ensure that all sensors within a zone are in range with each other.

Table 1 shows the mean energy consumed per sensor per message for various message encoding formats. As expected, we see that using ASCII based XML format, both the transmission and reception energy consumed is almost 9 times higher than that of custom encoding approach. On the other hand the use of a compact binary XML encoding format (with predetermined, separated XML schema) consumes same amount of energy as that of a custom encoding format. Completely customized protocols are most energy efficient but offer very little flexibility and interoperability. Whereas the textual XML represents the other end of the spectrum. Using XML has a number of attractive benefits, but the verbosity incurs large energy costs during communication (almost 9 times higher than the custom based formats). However, the use of binary XML is a compromise between textual XML and custom formats. The results helped us to validate our simulation framework.

## 6 Related Work

In an open distributed system, the problem of naming and locating resources is challenging and has been well studied in literature [1, 10]. Several protocols for service discovery have been proposed [13, 23, 25]. Moreover, in mobile environments, the effect of mobility has been also considered (e.g., [4, 17]).

Jini [13] supports service discovery via service registration and lookup. While Jini works well for its intended applications, we believe that it is not ideal for realizing DRD for sensor networks. First, Jini is Java-based, and this permeates its design. In theory, one could bind its wire formats and protocols to other languages, but the result would likely be awkward. For example, Jini's service description model is based on Java's rules for class derivation. A service query

matches a service if it is a supertype or same type as the service. This categorically precludes the use of a number of promising research areas for wide-scale interoperability and mediation, such as ontologies, RDF query languages [18, 14], description logics [2], and semantic mediation [9]. Furthermore, its communication model is based on Java RMI and it relies heavily on passing Java objects across network. This can impose high energy costs for small objects, which will be typical for DRD. The required changes to Jini to make it energy efficient would render it incompatible with the original specification and other technologies built with Jini.

Universal Plug and Play (UPnP) architecture [23] also aims for zero-configuration. However, it has been designed for a different context. In UPnP, components are divided into control points and devices. Each device has a device service. When a device comes on-line, it broadcasts itself to control points on the network. Once a control point has discovered a device, it then takes control of the device and sends control messages to the device's service. UPnP is for a network to discover devices, rather than for a client to discover resources, and does not include any notion of clustering. It is also designed for resource rich devices and its complex architecture is not easy to deploy on resource constrained sensors.

Additionally, many existing resource discovery schemes including Jini [13], UPnP [23], and SDS [4] assume an underlying IP based networking infrastructure. They also rely on support for multicast and assume presence of a reliable transport layer protocol such as TCP. However, the communication models as well as protocol stack architecture for sensor networks is still evolving and it is not clear whether the final model will support an IP based communication infrastructure or employ a complex machinery such as TCP. Therefore we believe that the existing technologies are not directly applicable to sensor networks.

Approaches developed in the context of ubiquitous computing [1] and mobile computing [4] share some of our design goals in terms of ubiquitous interoperation and energy efficiency respectively. However, they do not take advantage of other properties of sensor devices and applications (such as data-centric operation, and different levels and modes of granularity) to realize their systems.

Recently, Stann and Heidemann proposed resource discovery optimizations for sensor networks [20]. In this work, a homogeneous sensor network is assumed (at least, in terms of administration and software). In addition, resource discovery is integrated with the directed diffusion model which is used for data collection [12]; queries are forwarded towards data sources simultaneously discovering and reserving required resources in a position to report required data. The target of this work is to optimize this flooding process by taking advantage of historical queries for similar resources. The scope of our work is wider (not just query forwarding) and is not tied to the directed diffusion model.

## 7 Concluding Remarks

Resource discovery is an important first step towards enabling interoperability of sensor networks, and eventually seamless integration among them (what we

call *Virtual Sensor Networks*). In this paper, we define the resource discovery problem in sensor networks and outline the challenges involved in it. Sensor networks are unique in several respects that influence resource discovery and necessitate specialized solutions. More specifically, their data-centric embedded nature, relatively poor resources, the emphasis on energy efficiency and the lack of existing standards combine to render traditional resource discovery approaches ineffective.

The paper first explores the space of the resource discovery problem in sensor networks. Resource discovery was organized into two complementary components: the decision on what attributes of the system to track; and the design of energy efficient and available resource discovery protocol. We outlined the challenges involved in each of the subsections. In addition, we demonstrated a solution to one of the challenges (the balance between interoperability and efficiency in the resource discovery protocol) and showed that its possible to improve efficiency while maintaining interoperability.

We are also exploring the concept of Virtual Sensor Networks in the context of second generation sensor networks. More specifically, we are studying various challenges associated with them. To that end, we would like to propose a component-based, modular, efficient service-oriented architecture.

## References

1. W. Adjie-Winoto, E. Schwartz, H. Balakrishnan, and J. Lilley. The design and implementation of an intentional naming system. In *Symposium on Operating Systems Principles*, pages 186–201, 1999.
2. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider.
3. N. Bulusu, J. Heidemann, and D. Estrin. GPS-less low cost outdoor localization for very small devices. *IEEE Personal Communications Magazine*, Oct. 2000.
4. S. E. Czerwinski, B. Y. Zhao, T. D. Hodes, A. D. Joseph, and R. H. Katz. An architecture for a secure service discovery service. In *Mobile Computing and Networking*, pages 24–35, 1999.
5. Data Format Description Language project web page.  
<http://forge.gridforum.org/projects/dfdl-wg/>, 2004.
6. E. Christensen et. al. Web Services Description Language (WSDL) 1.1.  
<http://www.w3.org/TR/wsdl>, 2001.
7. G. Eisenhauer and L. K. Daley. Fast heterogenous binary data interchange. In *Proceedings of the Heterogeneous Computing Workshop (HCW2000)*, 2000.
8. R. Fikes, P. Hayes, and I. Horrocks. OWL-QL: A language for deductive query answering on the semantic web.  
<ftp://ftp.ksl.stanford.edu/pub/KSL-Reports/KSL-03-14.pdf.gz>, 2003. KSL Technical Report 03-14.
9. A. Gupta et al. Registering Scientific Information Sources for Semantic Mediation. In *21st International Conference on Conceptual Modeling*, 2002.
10. M. Harcol-Balter, P. Leighton, and D. Lewin. Resource discovery in distributed networks. In *Proc. of ACM PODS 1999*, pages 229–237, 1999.
11. W. Heinzelman. *Application-Specific Protocol Architectures for Wireless Networks*. PhD thesis, Massachusetts Institute of Technology, 2000.

12. C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proc. 6th ACM International Conference on Mobile Computing and Networking (Mobicom'00)*, Aug. 2000.
13. Sun microsystems. jini technology architectural overview (white paper). <http://www.sun.com/software/jini/whitepapers/architecture.html>.
14. G. Karvounarakis, S. Alexaki, V. Christophides, D. Plexousakis, and M. Scholl. Rql: a declarative query language for rdf. In *WWW '02: Proceedings of the 11th international conference on World Wide Web*, pages 592–603, New York, NY, USA, 2002. ACM Press.
15. P. Mockapetris and K. J. Dunlap. Development of the domain name system. In *SIGCOMM '88: Symposium proceedings on Communications architectures and protocols*, pages 123–133, New York, NY, USA, 1988. ACM Press.
16. Network Simulator. <http://isi.edu/nsnam/ns>.
17. C. Perkins and H. Harjono. Resource discovery protocol for mobile computing. *Mobile Networks Journal*, 1(4):447–455, 1996.
18. A. Seaborne. Rdql - a query language for rdf. <http://www.w3.org/Submission/2004/SUBM-RDQL-20040109/>, 2004.
19. Sqlite. <http://sqlite.org/>.
20. F. Stann and J. Heidemann. BARD: Bayesian-assisted resource discovery in sensor networks. In *Proceedings of the IEEE Infocom*, 2005.
21. Tinydb: In-network query processing in tinyos. <http://telegraph.cs.berkeley.edu/tinydb/doc/index.html>.
22. Tinyxml. <http://sourceforge.net/projects/tinyxml/>.
23. Universal plug and play device architecture. [http://www.upnp.org/download/UPnPDA10\\_20000613.htm](http://www.upnp.org/download/UPnPDA10_20000613.htm).
24. D. Veillard. Libxml2 project web page. <http://xmlsoft.org/>, 2004.
25. J. Veizades, E. Guttman, C. Perkins, and S. Kaplan. Service location protocol, 1997.
26. W3C. Simple Object Access Protocol (SOAP) 1.1. <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>, 2000.
27. W3C. Resource description framework (rdf). <http://www.w3.org/RDF/>, 2004.
28. Xerces: Xml parsers. <http://xml.apache.org/#xerces>.
29. Xml pull parser. <http://www.extreme.indiana.edu/xgws/xsoap/xpp/>.
30. Y. Xu, J. Heidemann, and D. Estrin. Geography-informed energy conservation for ad hoc routing. In *Proceedings of the 7th annual international conference on Mobile computing and networking*. ACM Press, 2001.

# Survey on Location Authentication Protocols and Spatial-Temporal Attestation Services

A.I. González-Tablas<sup>1</sup>, K. Kursawe<sup>2</sup>, B. Ramos<sup>1</sup>, and A. Ribagorda<sup>1</sup>

<sup>1</sup> Universidad Carlos III, Madrid, Spain  
Computer Science Department, SeTI  
{aigonzal, benja1, arturo}@inf.uc3m.es

<sup>2</sup> Katholieke Universiteit Leuven,  
Department Electrical Engineering, ESAT, COSIC  
klaus.kursawe@esat.kuleuven.be

**Abstract.** A survey on location authentication protocols and spatial-temporal attestation services is presented. Several protocols and services with these objectives have been proposed during the last decade, but still there is a lack of understanding of the security properties they should provide and which security mechanisms are appropriate. We first define the goals and threat model of location authentication protocols, next they are described and analyzed against this model. Also, spatial-temporal attestation services are described and classified depending on their goal and kind of issued evidence.

## 1 Introduction

The development of location technologies and the increasing mobility of our communications have allowed the deployment of Location Based Services (LBS). In this context some applications will benefit of authenticating the location of certain entity (*location authentication*) while others would prefer to obtain an evidence about the spatial-temporal conditions of certain entity or document (*spatial-temporal attestation*). For example, a service provider may require that in order to grant access to a service, their clients must be located at some specific set of locations, or a shopping center may desire to grant privileges depending on the visiting history to the center. In another context, spatial-temporal attestation services can be used to notarize from where some data is being sent, where some document is signed or where a certain payment is done. Another application is to provide accountability to the tracking of entities or assets. During the last 10 years several location authentication protocols and spatial-temporal attestation services have been proposed. Still there is a lack of understanding of the security properties they should provide and which security mechanisms are appropriate. In this paper these issues are addressed: a comprehensive survey on these protocols and services is presented and at the same time its security is analyzed.

## 2 Location Authentication Protocols

### 2.1 Definitions, Assumptions and Threat Model

The general setting for a location authentication protocol involves a *prover* ( $P$ ) and a *verifier* of the location ( $V_{loc}$ ).  $P$  is an entity which has some means for being located by a *positioning infrastructure*  $PI$  (see [HB01] for a survey on location systems) and that we assume to have an unique identification  $p$ . The verifier  $V_{loc}$  is presented with, or presumes beforehand, the purported location of the prover. Then, **location authentication** is defined as *the process whereby one party ( $V_{loc}$ ) is assured (through acquisition of corroborative evidence) of the location of a second party ( $P$ ) in a protocol, and that the second party has actually participated in the protocol (i.e., is active at, or immediately prior to, the time the evidence is acquired)*. A set of *locating entities*  $LE$ , which are part of the positioning infrastructure, may collaborate with  $V_{loc}$  to authenticate the prover's location.  $V_{loc}$  may be also part of the positioning infrastructure. We define that a **location authentication protocol is secure** if *in all its executions run with an adversary  $\mathcal{A}$  present,  $V_{loc}$  accepts the claim that the prover  $p$  is in location  $l$  at time  $t$  iff this statement is true*. The goal of an adversary  $\mathcal{A}$  is that  $V_{loc}$  accepts claims on target tuples  $\tau_t = (p_t, l_t, t_t)$  such that some or several of the elements of the tuple makes the statement ' $p_t$  was in location  $l_t$  at time  $t_t$ ' false.

We assume that provers are physical devices which know a secret  $s$  that allows to prove its identity  $p$  to other entities. This secret  $s$  is stored in a tamper-resistant module such that all the operations that use  $s$  are done inside this module and  $s$  cannot be leaked. However, the adversary  $\mathcal{A}$  can manipulate other  $P$ 's physical characteristics in order to subvert the protocol. Authenticating  $P$ 's location does not provide guarantees about who is the user  $U$  that may be controlling it. Although some mechanisms to authenticate the proximity of the user to the device can be used, such as protecting  $s$  with some other secret known by  $U$  or using biometric methods, we assume that both are bound to each other.

We assume that the adversary  $\mathcal{A}$  has under his control a set of compromised provers  $\mathcal{P}^* = \{p_1^*, \dots, p_n^*\}$ . The adversary can place these compromised provers in any location  $l \in \mathcal{L}$  chosen by the adversary at any time  $t \in \mathcal{T}$  and make them to execute a location authentication protocol with  $V_{loc}$ , to communicate securely between them using radio, sound or other mediums, or to capture, intercept or insert any message. Once an execution of a protocol has started,  $\mathcal{A}$  cannot move the compromised provers arbitrarily at his will if this movement is against the physic laws, but he can force them to not follow the steps of the protocol or to claim a different identity.  $\mathcal{A}$  can also record executions of the protocol run by provers under his control or by other provers, and use this information in later executions.  $\mathcal{A}$  may also want to know the whereabouts of provers which are not under his control, that is,  $\mathcal{A}$  would like to attempt against the privacy of provers  $p \notin \mathcal{P}^*$ . We are not going to analyze the protocols against this threat. In the same way, we will not consider denial of service attacks, even when it is very easy to run them successfully in the depicted scenario (e.g. jamming).

- 
- A) **Initialization.**
1.  $V_{loc}$  generates uniformly at random  $k$  bits  $\alpha_i$ .
  2.  $P$  generates uniformly at random  $k$  bits  $m_i$ .
- B) **Commitment.**
1.  $P$  commits to the  $k$  bits  $m_i$  using a secure commitment scheme protocol.
- C) **Fast exchange.** This phase is run repeatedly  $k$  times for  $i = 1 \dots k$ :
1.  $V_{loc}$  starts a timer.
  2.  $V_{loc} \rightarrow P : \alpha_i$
  3.  $P \rightarrow V_{loc} : \beta_i = \alpha_i \oplus m_i$  (immediately after it receives  $\alpha_i$ )
  4.  $V_{loc}$  stops the timer and measures the latency time  $\lambda_i$ .
- D) **Commitment opening.**
1.  $P$  opens its commitments on bits  $m_i$  to  $V_{loc}$ .
- E) **Authentication and verification.**
1.  $P$  builds  $m = \alpha_1|\beta_1|\dots|\alpha_k|\beta_k$ , signs this value  $m$  and sends the result to  $V_{loc}$ .
  2.  $V_{loc}$  verifies if the committed bits in step B.1 are the same as  $\alpha_i \oplus \beta_i$ . If this holds,  $V_{loc}$  computes  $m$  as  $P$  would have done and verifies  $P$ 's signature on  $m$ . If this also holds,  $V_{loc}$  computes an upper-bound on the distance using the maximum of the measured latency times  $\max(\lambda_i)$  with  $i = 1, \dots, k$ , and accepts if and only if  $P$  is close by.
- 

**Fig. 1.** Brands-Chaum distance-bounding protocol [BC94]

## 2.2 Distance-Bounding Protocols

The goal of these protocols is to authenticate that the prover  $P$  is within some distance  $d_{lim}$  from some location  $l_0$  where a locating entity  $LE$  or a verifier  $V_{loc}$  is placed. Without losing generality, the set of locations  $\mathcal{L}_t$  that the adversary may target is defined as  $\mathcal{L}_t = \{l_t : l_t \in \mathcal{L}, d(l_0, l_t) \leq d_{lim}\}$ , where  $d(\cdot, \cdot)$  is a function that returns the distance between two locations. Following we describe and analyze the distance-bounding protocols that currently exist in the literature.

**Based on Fast Challenge-Response Exchanges.** Some distance-bounding protocols are based on the measurement of the round trip latency  $\lambda$  between  $P$  and  $V_{loc}$ . They are designed as interactive two-party protocols and the main assumption is that the signals used to transmit the exchanged messages have a constant propagation speed  $v$ , where  $v = v_c \cong 3 \times 10^8 m/s$  if radio or optical signals are used and  $v = v_s \cong 340 m/s$  if sound. The round trip latency is defined as  $\lambda = t_{pp}(l_0, f(P, t_{run})) + t_{pc}(P) + t_{pp}(f(P, t_{run}), l_0)$ , where  $t_{pp}(l_1, l_2)$  is the propagation time between location  $l_1$  and location  $l_2$ ,  $t_{pc}(P)$  the processing time of a prover  $P$  between the reception of a challenge and the transmission of its response and function  $f(p, t)$  returns the location of prover  $p$  at time  $t$ .

Brands and Chaum were the first that proposed a protocol falling in this category in [BC94] (see Figure 1). Their protocol and the ones in [ČBH03, Bus04] assume that the device has some hardware that performs the exchange in a fast manner over some dedicated communication channel. Then, they assume that the prover's processing time is negligible compared to the propagation time and an upper-bound of the distance between  $V_{loc}$  and  $P$  can be computed as  $\delta = v \times \lambda/2 \geq d(l_0, f(p, t_{run}))$ . Other proposals in [SSW03, WF03] assume that the device has a non-zero processing delay. Then, the upper-bound is calculated as  $\delta = v \times (\lambda - t_{pc}(P))/2$ . In [SSW03], responses are sent using sound while challenges use radio signals, then  $\delta \cong v \times (\lambda - t_{pc}(P))$ .

Assuming that the adversary  $\mathcal{A}$  controls a single prover  $p_i^*$  such that  $f(p_i^*, t_t) = l_i^* \in \mathcal{L}_t$ , then  $\mathcal{A}$  may try to impersonate some prover  $p_t = p_j \neq p_i^*$  (*impersonation attack*). This would be possible if provers were not authenticated at any time during the execution of the protocol. Most of the distance-bounding protocols based on fast exchanges authenticate provers. On the contrary, in the protocol in [SSW03] provers are not authenticated (it is not considered a goal), therefore the impersonation attack does not make any sense. The protocol in [WF03] does not authenticate the prover, but the whole spatial-temporal certification protocol which uses it in a phase does, preventing then this attack.

With a single compromised prover  $p_t = p_i^*$  such that  $l_i^* \notin \mathcal{L}_t$ ,  $\mathcal{A}$  may try to decrease the measured latency  $\lambda$  (*decreasing measured latency attack*) with respect to the one that should have been measured (note that trying to increase  $\lambda$  will not help  $\mathcal{A}$  to get the claim accepted). First,  $\mathcal{A}$  may try to send the response in advance of receiving the challenge from  $V_{loc}$ . To avoid this, the response in this kind of protocols is chosen such as it depends on the challenge and a value which  $P$  commits to previously, as in the protocol in Figure 1. If it can be assumed that the propagation speed of the signals used to exchange the messages has an upper-bound which no prover can exceed, including those controlled by  $\mathcal{A}$ , then the probability for  $\mathcal{A}$  guessing a response  $r \in \{0, 1\}^m$ , and succeeding in the attack, is  $1/2^m$ . To increase the security of the protocol, several exchanges can be done. In the case of the proposal in [SSW03] the previous assumption does not hold, because the response is transmitted using sound. Then  $\mathcal{A}$  may try to decrease  $\lambda$  by using a faster signal in some part of the trajectory.

In protocols in [SSW03, WF03] a non-zero processing time is assumed, then  $\mathcal{A}$  may try to decrease the measured latency  $\lambda$  by decreasing this time. To avoid this, in [WF03] it is proposed that this time is known by the verifiers and it is assumed that  $\mathcal{A}$  cannot manipulate it. On the contrary, in [SSW03] it is assumed that  $\mathcal{A}$  may tamper this time; an effective countermeasure is proposed based on decreasing  $d_{lim}$  dynamically depending on the  $P$ 's declared processing time:  $d_{lim}(t_{pc}(P)) = d_{lim}(0) - t_{pc}(P) \times v$ .

If  $\mathcal{A}$  controls a single prover  $p_i^*$ , he may try the attack referred as *mafia fraud* in [BC94, Bus04] or *proxy attack* in [WF03]. Prover  $p_i^*$  impersonates  $V_{loc}$  in order that  $p_t = p_k \notin \mathcal{P}^*$  run the protocol with  $p_i^*$  instead of with  $V_{loc}$ . It is assumed that  $d(l_0, f(p_k, t_t)) > d_{lim}$  and  $d(l_0, l_i^*) \leq d_{lim}$ . The protocols in [BC94, WF03, ČBH03, Bus04] prevent this attack as the distance between  $p_k$  and  $p_i^*$  makes  $\lambda$  increase and  $V_{loc}$  will not accept the claim (assuming that the propagation speed has an upper-bound which cannot be exceeded). The protocol in [SSW03] would prevent this attack if  $\mathcal{A}$  could not use signals which propagate faster than sound, but this is not assumed in the protocol. Anyhow, as in [SSW03] anyone can impersonate other provers, the proxy attack does not make any sense.

When the adversary controls at least two provers  $p_i^*$  and  $p_j^*$ , which is a reasonable scenario, then the attack referred as *collaborator attack* in [BC94] or *terrorist attack* in [Bus04] can take place. Then the target tuple  $\tau_t = (p_i^*, l_t, t_t)$  is such that  $l_i^* \notin \mathcal{L}_t$  but  $l_j^* \in \mathcal{L}_t$ . If the fast exchange phase is not bound to the identity of the prover that executes the protocol, it can be done by a dif-



- 
1.  $P \rightarrow V_{loc} : P, R, L$
  2.  $V_{loc} \rightarrow LE : K_s \{P, N\}, K_{V_{loc}, LE} \{K_s\}$
  3.  $LE \xrightarrow{rc\lambda^L} P : P, N$
  4.  $P \rightarrow V_{loc} : P, R, N$
  5.  $V_{loc}$  verifies that the token  $N$  received in step 4 is the same as the one it sent to  $LE$  in step 2.
- 

**Fig. 2.** Kindberg-Zhang distance-bounding protocol [KZ01b]

ferent one. For example, in protocol in Figure 1  $p_j^*$  may sit between  $p_i^*$  and  $V_{loc}$  and act as a transparent proxy between them in all the phases except in the Phase C, which  $p_j^*$  will execute by itself if  $p_i^*$  has already communicated  $p_j^*$  the bits  $m_i$ . Protocols in [BC94, WF03, SSW03, ČBH03] are vulnerable to this attack (however note that this attack does not make any sense in [SSW03] again). The proposal in [Bus04] solves this problem by binding the secret  $s$  to the fast exchange phase. In Bussard’s protocol the response depends also on  $s$  in such a way that this dependency can be proved without revealing  $s$  by using proof of knowledge protocols. This protocol is secure (with some probability) if it can be assumed that the signals’ speed has an upper-bound that cannot be exceeded.

**Based on Token Broadcast.** Other distance-bounding protocols are based on broadcasting some token  $N$  through a set of short-range beacons playing the role of  $LE$ . Protocols proposed in [KZ01b, Mic03] are of this kind. In this setting it is assumed that the token can only be received if  $d(l_0, f(p, t_{run})) < d_{lim}$ ,  $d_{lim}$  determining the end of  $LE$ ’s transmission range. Then, knowing  $N$  is assumed to be a proof of having been close to  $LE$ . As Kindberg and Zhang discuss in [KZ01b] this assumption can be reasonably held in certain scenarios (e.g., if infrared or ultrasound signals are used and the region is delimited with walls).

If the adversary controls a single prover  $p_i^*$  such that  $d(l_0, l_i^*) > d_{lim}$ , he may try to guess  $N$  (*guessing attack*). To prevent this attack, tokens should be random nonces, and at the same time depend on the area and the broadcast time to prevent *reuse attacks*. Protocols in [KZ01b, Mic03] prevent these attacks.

As in the previous setting,  $\mathcal{A}$  may try to perform *impersonation attacks*, to prevent them, some kind of prover authentication is needed. However, protocols falling in this category do not agree with this approach. Protocol in [Mic03] does not authenticate provers during execution (one of its main goals is prover anonymity). Kindberg and Zhang in [KZ01b] claim that entity authentication or anonymity issues are orthogonal to the location authentication problem, and therefore they do not consider this issue in their protocols (see protocol in Figure 2). As in our model impersonation attacks are relevant, prover authentication should be required.

*Proxy attacks* may also be run in this setting. In a first version  $\mathcal{A}$  will target a tuple  $\tau_t = (p_i^*, l_t, t_t)$  such that  $d(l_0, l_i^*) > d_{lim}$ . The attack involves a prover  $p_k \notin \mathcal{P}^*$  such that  $d(l_0, l_k) \leq d_{lim}$  and  $p_i^*$  sits between  $p_k$  and  $V_{loc}$ , acting as a transparent proxy between them and playing the role of  $V_{loc}$  to  $p_k$ . The protocol in [Mic03] is vulnerable to this attack as it does not authenticate provers. Protocol in [KZ01b] could prevent this attack if  $V_{loc}$  authenticated provers and

kept a registry binding broadcast tokens with specific prover requests, or if this binding were done within the token and its authenticity preserved (assuming that honest provers would not accept tokens not addressed to them). A second version of the proxy attack is that one where the prover  $p_i^*$  sits between  $LE$  and  $p_k$ , acting again as a transparent rely between them.  $\mathcal{A}$  targets the tuple  $\tau_t = (p_k, l_t, t_t)$  where  $d(l_0, l_i^*) \leq d_{lim}$  but  $d(l_0, l_k) > d_{lim}$ . This attack would not be detected even if tokens were bound to provers in an authentic manner. A possible countermeasure suggested in [KZ01b] is that  $V_{loc}$  measures the response time to verify if it corresponds to the expected distance between  $V_{loc}$  and  $p_k$ ; then similar techniques to the ones presented in the previous section will be applied. Another countermeasure might be that  $LE$  used unforgeable RFID schemes and that tokens were bound to each detected prover.

If  $\mathcal{A}$  controls at least two provers  $p_i^*$  and  $p_j^*$  then *collaborator attacks* may take place. Target tuple would be  $\tau_t = (p_i^*, l_t, t_t)$  where  $d(l_0, l_i^*) > d_{lim}$  and  $d(l_0, l_j^*) \leq d_{lim}$ . All the considerations made before for proxy attacks are relevant, but in this case  $p_j^*$  will collaborate in the attack (e.g. accepting tokens not addressed to it). Again, a possible countermeasure would be that  $LE$  authenticates provers in the acceptance area and binds tokens to them.

### 2.3 Absolute Positioning Protocols

The goal of these protocols is to authenticate  $P$ 's absolute position with some resolution. These protocols usually rely on triangulation techniques. If the target prover is  $p_t = p_i^*$  such that  $f(p_i^*, t_t) = l_i^*$ , then  $\mathcal{L}_t = \{l_t : l_t \in \mathcal{L}, l_t \neq l_i^*\}$ . Following the two kind of protocols falling in this category are described.

**Based on Simultaneous Fast Challenge-Response Exchanges.** As these protocols are designed as the simultaneous execution of several distance-bounding protocols based on fast exchanges run by  $P$  and several  $LE$ , then, the analysis presented in the previous section for distance-bounding protocols based on fast exchanges can be applied to this setting. Given this, let's assume that  $\mathcal{A}$  controls one single prover  $p_i^*$ , that the speed of the exchanged signals cannot be exceeded by any prover and that some countermeasures have been applied to prevent manipulation of processing times if devices are assumed to have any. Then  $\mathcal{A}$  may try to prove another location  $l_t \neq l_i^*$  by delaying prover's answers. Čapkun and Hubaux prove in [ČH04] that if the prover lies within the triangle with vertices each  $LE$ , it cannot prove successfully being at another location than where it actually is. A prover can always prove to be further from one of the  $LE$  but then, if it lies within the mentioned triangle, it must prove to be closer to at least other of the  $LE$ , which is not possible under the assumptions.

**Based on Authenticated Ranging.** Some location authentication protocols are based on signals broadcast by global navigation satellite systems (GNSS) such as GPS. In these systems several satellites orbiting around the earth transmit continuously signals  $L_i$ , where satellites play the role of  $LE$ . The positioning principle is based on measuring the time of flight from a satellite  $LE_i$  to the

prover  $P$ , which allows to compute their range or distance; several ranges can be used to calculate  $P$ 's absolute position by triangulation. This method needs that satellite and receiver clocks are synchronized, but usually there exists some bias or offset in the receiver clock respect to system time (satellite clocks are much more stable and precise); therefore to calculate the prover's position (latitude, longitude, height) the bias must be solved and at least four measurements are needed. In this section it is assumed that provers are GNSS receivers with added functionalities such as communication capabilities.

A first approach to authenticate  $P$ 's location at time  $t$  would be that the device calculates its position  $f(p, t)$  using the received navigation signals and sends a spatial-temporal report containing the tuple  $(p, f(p, t), t)$  to  $V_{loc}$ . If these reports are not protected,  $\mathcal{A}$  can intercept them and send faked ones instead (*report manipulation attack*). To avoid this, message authentication should be provided as it is suggested in [GW99, PWK04].

Even if reports were authenticated,  $\mathcal{A}$  might try to manipulate provers in order to transmit false reports (*device manipulation attack*). If  $\mathcal{A}$  controls one prover  $p_i^*$  located in  $l_i^*$  at time  $t_i^*$ , he may force  $p_i^*$  to send forged reports  $\tau_t = (p_t, l_t, t_t)$  such that  $p_t \neq p_i^*$ ,  $l_t \neq l_i^*$ ,  $t_t \neq t_i^*$  (if reports can be sent at a later time  $t_j > t_i^*$ ) or a combination of these. To avoid this threat in [PWK04] it is proposed to use tamper resistant receivers such that they only output authenticated spatial-temporal reports calculated with received navigation signals, and which can check its integrity status and send reports on it.

Anyway, even if these assumptions can be held,  $\mathcal{A}$  does not need to tamper provers to make them generate false reports. This is possible because satellite signals can be easily synthesized or manipulated with the appropriate software and fed to the device (*signal manipulation attack*). Anyhow, the price of these simulators or its hiring is high and in several applications it may not be worth compared to the benefit that  $\mathcal{A}$  would obtain. To avoid these attacks the authentication of the broadcast signals should be guaranteed. One approach is to use symmetric encryption, as it is done in one of the GPS signals, where spreading code encryption with a symmetric secret key is used. Other approach considers that satellites broadcast some unpredictable information which will be recorded by  $P$  and forwarded to  $V_{loc}$ . This approach is somehow used in [MMZ<sup>+</sup>97], where small errors such as satellite orbit errors and ionospheric errors are used as unpredictable information. However, Kuhn points out in [Kuh04] that with this mechanism anyone able to verify the correction of the unpredictable information could also spoof the signal by including this information on a synthesized signal or transforming the signal according to it; further research should be done to check if this kind of attack could be detected in this case. A last approach to provide authentication to broadcast signals is based on asymmetric cryptography. For example, the proposal by Kuhn in [Kuh04] uses digital signatures to provide protection against signal synthesis attacks and also selective delay attacks; in this case undetectable hidden markers are inserted in the signal at unpredictable times and, after some time, signed information that allows markers verification is broadcast.

$\mathcal{A}$  may try to run in this setting both variants of *proxy attacks*. The first version (where  $l_i^* \neq l_t$ ) is not possible in [PWK04] as devices are assumed to output authenticated reports, but the second version of the attack would make the device to calculate a wrong position unless it could detect that the signals had been forwarded. In [MMZ<sup>+</sup>97] the first version of the attack might be prevented if latency measurements or similar countermeasures were carried out, because reports are apparently not bound to a specific receiver. The second version of the attack would not easily succeed as devices must prove to be at a fixed set of positions. The forwarding of the signals and its feeding to the device will make  $V_{loc}$  to fail in the calculation of its position with a high probability. This last situation would happen also in [MMZ<sup>+</sup>97] if  $\mathcal{A}$  tried to carry out a *collaborator attack*, which would not be possible in [PWK04] as trusted devices are assumed.

### 3 Spatial-Temporal Attestation Services

Similar to the definition for non-repudiation services in [ISO97], we define **spatial-temporal attestation services** as *those services that generate, collect, maintain, make available and validate evidences concerning either the spatial-temporal conditions of an entity either the spatial-temporal conditions under which a transformation or action is made by some entity on certain data*. A trusted third party (TTP), the *spatial-temporal evidence generator* ( $G_e$ ), is in charge of generating the evidences, and probably also collects, maintains and makes them available. Another TTP may exist, the *spatial-temporal evidence verifier* ( $V_e$ ), if evidences cannot be verified by any party. We assume that the generator of the evidence  $G_e$ , before certifying the spatial-temporal conditions of the *subject* of the evidence, delegates the verification of these conditions to some entity  $V_{loc}$ , which should execute a location authentication protocol.

Assuming that spatial-temporal attestation services rely on secure location authentication protocols, the main goal of a spatial-temporal evidence generator  $G_e$  is to provide unforgeable, non-transferable and verifiable spatial-temporal evidences on tuples  $\tau = (p, l, t)$  such that it is true that the subject  $p$  was in location  $l$  at time  $t$ . The goal of an adversary  $\mathcal{A}$  is to obtain evidences on tuples  $\tau_t = (p_t, l_t, t_t)$  such that one or several of the elements of the tuple makes the statement ‘ $p_t$  was in location  $l_t$  at time  $t_t$ ’ not true.

Following, a classification of spatial-temporal attestation services is presented depending on their specific goal and which kind of evidence they issue. Most of the spatial-temporal attestation services existing in the literature use well known evidence generation mechanisms such as digital signatures, secure seals or authenticator tokens. Therefore, a security analysis as the one developed in Section 2 is not presented in this case.

**Spatial-temporal certification services.** A first kind of spatial-temporal attestation services are those that have as main goal to provide evidences on the spatial-temporal conditions of a subject. A first group between these services provide evidences that may be certificate-like or credential-like. The proposals

in [ZKK01, WF03] fall within the certificate-like category while the one proposed in [Bus04] can be classified as credential-like.

Other authors in [GTRR03] suggest to link certificate-like spatial-temporal evidences, as it is done in linked time-stamps schemes, to provide accountability to the temporal order of the evidences. Some of the protocols presented in [CBH03] also provide some kind of spatial-temporal evidence but the location is not explicitly included in the evidence, they are more like temporal authenticators of the encounters between entities than proper spatial-temporal evidences.

A second group between spatial-temporal certification services provides ticket-like evidences or short-term credentials, such as the protocol in [Mic03]. Another protocol that falls into this approach is presented in [NNT03], but in this case the ticket, which is more similar to an authenticator than to a proper credential, can be used only a limited number of times.

**Spatial-temporal stamping services.** A second kind of spatial-temporal attestation services are those that have as main goal to provide evidences about the spatial-temporal conditions under which some document exist or a transformation is made by some subject on this document. In this case,  $G_e$  may issue spatial-temporal stamps, which bind the document or its transformation with the spatial-temporal conditions. One of the more interesting transformations is to sign some data, which can be useful for example if some payment is done or some contract or attestation is signed. The only proposals that really fall under this approach are the ones presented in [KZ01a, LSBP03].

## 4 Conclusions and Open Issues

The expectations raised by recently proposed location authentication protocols and spatial-temporal attestation services are very promising. Although several protocols and services with these objectives have been proposed in the last decade, there is a lack of a framework that comprises them and that helps to analyze its security. In this paper we have surveyed existing location authentication protocols and spatial-temporal attestation services, their goals have been stated and its security has been analyzed against a proposed threat model.

There are still some open issues that should be further studied such as to analyze the efficiency of the protocols and services, the privacy they provide or how they may defend against denial of service attacks. The results of this work may be applied to analyze the security of the standardized positioning techniques in the context of mobile telephone networks.

## Acknowledgment

The authors would like to thank the anonymous referees for their useful comments and suggestions. First author wants to thank Karel Wouters from K.U. Leuven for sharing several discussions about the collaborator scenario. First, third and fourth authors are partly supported by “Dirección General de Investigación del M.E.C.” under contract SEG2004-02604: ‘*CERTILOC: Digital CERTification service for LOCation information*’.

## References

- [BC94] S. Brands and D. Chaum. Distance-bounding protocols. In *Workshop on the theory and application of cryptographic techniques on Advances in cryptology*, pages 344–359. Springer-Verlag New York, Inc., 1994.
- [Bus04] L. Bussard. *Trust Establishment Protocols for Communicating Devices*. PhD thesis, Institut Eurécom, Télécom Paris, 2004.
- [ČBH03] S. Čapkun, L. Buttyán, and J. P. Hubaux. SECTOR: Secure Tracking of Node Encounters in Multi-hop Wireless Networks. In *1st ACM Workshop on Security in Ad Hoc and Sensor Networks*, October 31, 2003.
- [ČH04] S. Čapkun and J. P. Hubaux. Securing position and distance verification in wireless networks. Technical Report EPFL/IC/200443, EPFL, May 2004.
- [GTRR03] A. I. González-Tablas, B. Ramos, and A. Ribagorda. Path-Stamps: A proposal for enhancing the security of location tracking applications. In *Ubiquitous Mobile Information and Collaboration Systems Workshop*, 2003.
- [GW99] E. Gabber and A. Wool. On location-restricted services. *IEEE Network*, November/December 1999), 1999.
- [HB01] J. Hightower and G. Borriello. A Survey and Taxonomy of Location Systems for Ubiquitous Computing. Technical Report UW-CSE 01-08-03, University of Washington, 2001.
- [ISO97] ISO/IEC 10181-4. Information technology - OSI - Security frameworks in open systems - Part 4: Non-repudiation framework, 1997.
- [Kuh04] M. Kuhn. An asymmetric security mechanism for navigation signals. In *6th Information and Hiding Workshop*, 23-25 May 2004.
- [KZ01a] M. Kabatnik and A. Zugenmaier. Location stamps for digital signature: A new service for mobile telephone networks. In *Networking - ICN 2001, First International Conference*, LNCS 2094. Springer, 2001.
- [KZ01b] T. Kindberg and K. Zhang. Context authentication using constrained channels. Report HPL-2001-84. Technical report, HP Labs Tech., 2001.
- [LSBP03] A. Lakshminarayanan, V. Singh, F. Bao, and K. P. Prabhu. Patent WO 03/007542. Method for certifying location stamping for wireless transactions, 2003. Publication date: 23/01/2003.
- [Mic03] N. Michalakakis. Location aware access control for pervasive computing environments. Master's thesis, MIT, 2003.
- [MMZ<sup>+</sup>97] P. F. MacDoran, M. B. Mathews, F. A. Ziel, K. L. Gold, S. M. Anderson, M. A. Coffey, and D. E. Denning. Patent WO 97/13341. Method and Apparatus for Authenticating the Location of Remote Users of Network Computing Systems, 1997. Publication date: 10/04/1997.
- [NNT03] K. Nakanishi, J. Nakazawa, and H. Tokuda. LEXP: Preserving user privacy and certifying the location information. In *Proc. of the 2nd Workshop on Security in Ubiquitous Computing (UBICOMP 2003)*, October 2003.
- [PWK04] O. Pozzobon, C. Wullems, and K. Kubik. Secure tracking using Galileo services. In *Proc. of the 2004 Intl. Symposium on GNSS/GPS*, 2004.
- [SSW03] N. Sastry, U. Shankar, and D. Wagner. Secure verification of location claims. In *Proc. of the 2003 ACM workshop on Wireless security*. ACM Press, 2003.
- [WF03] B. R. Waters and E. W. Felten. Secure, Private Proofs of Location. TR-667-03. Technical report, Princeton, Computer Science, January 2003.
- [ZKK01] A. Zugenmaier, M. Kreutzer, and M. Kabatnik. Enhancing applications with approved location stamps. In *Proc. of IEEE Intelligent Network 2001 Workshop (IN2001)*, 2001.

# Predicate Detection Using Event Streams in Ubiquitous Environments

Ajay D. Kshemkalyani

Computer Science Department, Univ. of Illinois at Chicago, Chicago, IL 60607, USA  
ajayk@cs.uic.edu

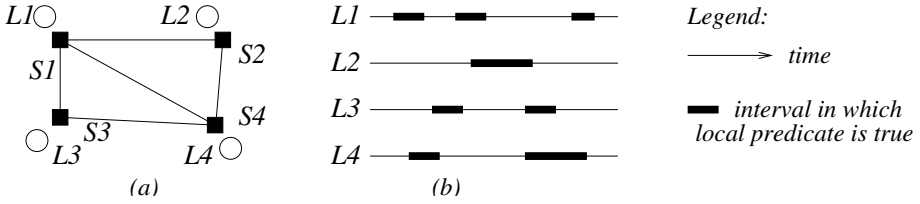
**Abstract.** Advances in clock synchronization techniques for sensor networks as well as wireless ad-hoc networks allow an approximated global time for an increasing number of configurations in ubiquitous and pervasive computing environments. This paper presents an event stream based on-line algorithm that fuses the data reported from the heterogeneous processors in the network to detect predicates of interest. The algorithm detects properties that can be specified using predicates under a rich palette of time modalities. The algorithm has low space, time, and message complexities. The main idea used to design the algorithm is that the predicate is decomposed as a collection of predicates between pairs of system devices. The algorithm leverages the *pairwise interaction* between processes so as to incur a low overhead and hence be highly scalable.

## 1 Introduction

Event-based data streams represent relevant state changes that occur at the processes that are monitored. The paradigm of analyzing event streams to mine data of interest to various applications uses *data fusion*. This paper gives an on-line algorithm to detect predicates from event streams that are reported by the various components of an ubiquitous computing environment. Such an environment includes ad-hoc networks and sensor networks [1, 19].

In the system model, the devices of the ubiquitous network are modeled by processes. The model assumes a loosely-coupled ad-hoc asynchronous message-passing system in which any two processes belonging to the process set  $N = \{P_1, P_2, \dots, P_n\}$  can communicate over logical channels. For a wireless communication system, a physical channel exists from  $P_i$  to  $P_j$  if and only if  $P_j$  is within  $P_i$ 's range; a logical channel is a sequence of physical channels representing a multi-hop path. The only requirement is that each process be able to send its gathered data eventually and asynchronously (via any routes) in a FIFO stream to a data fusion server  $P_0$ .

$E_i$  is the linearly ordered set of events executed by process  $P_i$  in an execution. Variable  $x$  local to process  $P_i$  is denoted as  $x_i$ . Given a network-wide predicate on the variables, the *intervals* of interest at each process are the durations during which the local predicate is true. Such an interval at process  $P_i$  is identified by the (totally ordered) corresponding adjacent events within  $E_i$ , for which the local predicate is true. Intervals are denoted by capitals  $X$ ,  $Y$ , and  $Z$ . The types



**Fig. 1.** Intervals within an ubiquitous network. (a) A network.  $S1 - S4$  are sensors at locations  $L1 - L4$ . (b) Timing diagram for intervals at  $L1 - L4$ .

of predicates our algorithm handles are *conjunctive* predicates. A *conjunctive* predicate is of the form  $\bigwedge_i \phi_i$ , where  $\phi_i$  is any predicate defined on variables local to process  $P_i$ . An example is:  $(x_i > 4) \wedge (y_j = 94)$ , where  $x_i$  and  $y_j$  are variables at  $P_i$  and  $P_j$ , respectively. Figure 1 shows four locations and intervals in their timing diagrams, during which the local predicates are true.

The problem we address is informally described as follows. Event streams from the processes report intervals in which the local predicates are true. Information about the reported intervals is “fused” or correlated and examined to *detect* global states of the execution that satisfy a given input predicate. This problem was defined and addressed earlier [3, 4, 5, 12] to detect predicates in distributed executions, using causality relationships defined in [11, 12].

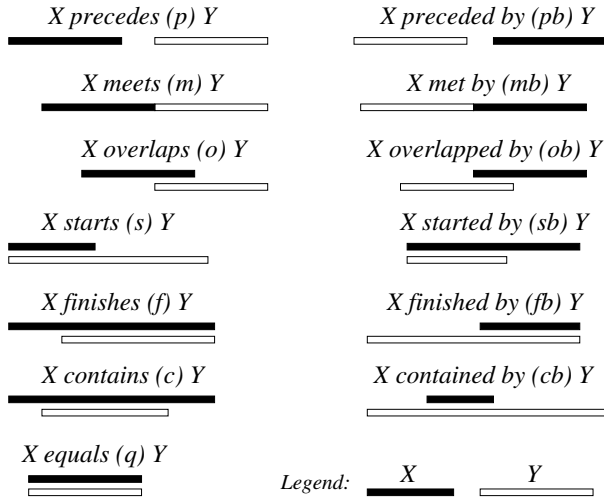
Physical clocks in sensor networks, ad-hoc networks, and wireless networks – when synchronized via GPS [15], NTP [16], or any of the many efficient synchronization protocols for wired as well as wireless media, such as those surveyed in [8, 9, 17, 18] – allow the assumption about an approximate single global time axis. We assume such synchronized physical clocks. This assumption simplifies the detection of a global state [7] in the ubiquitous environment, which is essentially a form of a distributed asynchronous message-passing system. With synchronized clocks, a *distributed execution* is the interleaving of all the local executions  $E_i$  on a common time axis. A *global state* contains one local state of each process. Using a common time axis, a global state can be specified (i) as occurring at the same time instant at each process, or (ii) in terms of specific relationships among the local states (one local state from each processes).

For a single time axis, it has been shown [10, 2] that there are 13 ways in which two time intervals can be related to one another on that time axis. For intervals  $X$  and  $Y$ , the thirteen relations are:

- *precedes* and *preceded by* (which is *precedes*<sup>-1</sup>)
- *meets* and *met by* (which is *meets*<sup>-1</sup>)
- *overlaps* and *overlapped by* (which is *overlaps*<sup>-1</sup>)
- *contains* and *contained by* (which is *contains*<sup>-1</sup>)
- *starts* and *started by* (which is *starts*<sup>-1</sup>)
- *finishes* and *finished by* (which is *finishes*<sup>-1</sup>)
- *equals*

The set of these 13 relations is denoted  $\mathfrak{R}$  and is illustrated in Figure 2. There are six pairs of inverses, and *equals* is its own inverse.





**Fig. 2.** The 13 relations  $\mathfrak{R}$  between intervals

Our problem is now formally defined. Event streams generated by the different processors need to be fused at a central server to solve the following global predicate detection problem [3, 4, 5, 6, 12].

**Problem Predicate\_Rel statement.** Given a relation  $r_{i,j}$  from  $\mathfrak{R}$  for each pair of processes  $P_i$  and  $P_j$ , identify the intervals (if they exist), one from each process, such that each relation  $r_{i,j}$  is satisfied for the  $(P_i, P_j)$  pair.

**Example specification:** We assume that intervals  $X_i, Y_j,$  and  $Z_k$  occur at different locations  $i, j,$  and  $k,$  respectively, but global time is available in the system at all sites. Two example specifications of predicates are:

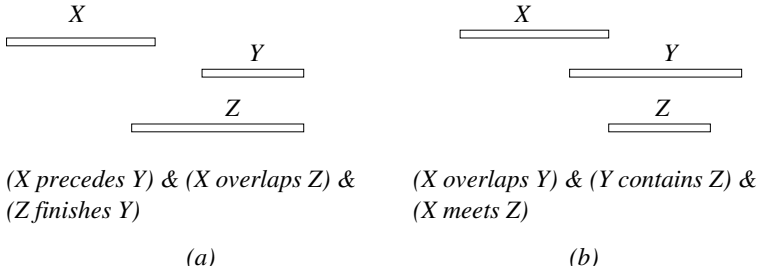
- (a)  $(X_i \text{ precedes } Y_j)$  AND  $(X_i \text{ overlaps } Z_k)$  AND  $(Z_k \text{ finishes } Y_j)$
- (b)  $(X_i \text{ overlaps } Y_j)$  AND  $(Y_j \text{ contains } Z_k)$  AND  $(Z_k \text{ met by } X_i)$

The problem in each case is to identify the global state in a distributed execution when the predicate is true. Example solutions are illustrated in Figure 3.

The performance of the proposed algorithm is summarized in Table 1.  $P_0$  is the data fusion server that processes the event streams. The metrics are the time complexity at  $P_0$ , the network-wide count of the messages sent by the processes to  $P_0$ , the total space complexity at  $P_0$ , the cumulative size of all the messages sent to  $P_0$ , and the space at each process  $P_i$ .  $n =$  number of processes,  $p =$  maximum number of intervals occurring at any process.

## 2 Outline of the Algorithm

An interval at  $P_i$  begins when the local predicate  $\phi_i$  becomes true and ends when  $\phi_i$  becomes false. We assume the physical clock has infinitely fine granularity so



**Fig. 3.** Example problem specifications. The intervals  $X_i$ ,  $Y_j$ , and  $Z_k$  are at different processes in the distributed ubiquitous system.

**Table 1.** Space, message and time complexities of the proposed algorithm

Time complexity at $P_0$	Total number of messages	Space at $P_0$ (=total message space)	Space at $P_i$ , $i \in [1, n]$
$O((n - 1)np)$	$np$	$2np$	2

each (event-triggered) state transition at a process occurs at a distinct tick (*local discreteness*). There are two consequences of *local discreteness* and the model for intervals. (1) An interval has a non-zero duration, implying *points* are not allowed. (2) An interval can begin at  $P_i$  only after the previous interval at  $P_i$  ends (see Fig. 1(b)) – termed the *local interval separation* property.

Processes  $P_1, P_2, \dots, P_n$  representing the  $n$  devices in the ubiquitous network track the start and end timestamps of their local intervals. The timestamps are as per the synchronized physical clock based on a global time axis.  $t_i^-$  and  $t_i^+$  denote the timestamps at process  $P_i$  at the start and at the end of an interval, respectively. This information is sent using the network asynchronously to the central data fusion server  $P_0$ . The only requirement is that between any process and  $P_0$ , the logical link must be FIFO. Recall that in a ubiquitous network, the numerous small devices operating collectively, rather than as stand-alone devices, form a dynamic ambient network that connects each device to more powerful networks and processing resources. Thus, the design using  $P_0$  as the more powerful “server” fits this model.

The data fusion server maintains queues  $Q_1, Q_2, \dots, Q_n$  for interval information from each of the processes. The server runs the proposed algorithm to process the interval information it receives in the queues. For any pair of intervals, observe from Figure 2 that there is an overhead of  $O(1)$  time complexity to test for each of the 13 relations using the start and end timestamps of the intervals. The algorithm detects “concurrent” pairwise interactions for each pair of intervals, considering only one interval from each process at a time as being a part of a potential solution. A challenge to solve *Predicate\_Rel* is to formulate the necessary and sufficient conditions to determine when to eliminate the received information about intervals from the queues, so as to process the queues

```

type Log = record      Start of an interval:      End of interval:
    start : integer;      Logi.start =  $t_i^-$ .      Logi.end =  $t_i^+$ 
    end : integer;      Send Logi to central process  $P_0$ .
end

```

**Fig. 4.** Data structures and operations to construct  $Log$  at  $P_i$  ( $1 \leq i \leq n$ )

efficiently. These conditions are important because intervals that are determined as not being part of a solution should not be used in testing with other intervals.

We assume that interval  $X$  occurs at  $P_i$  and interval  $Y$  occurs at  $P_j$ . For any two intervals  $X$  and  $X'$  that occur at the same process, if  $precedes(X, X')$ , then we say that  $X$  is a *predecessor* of  $X'$  and  $X'$  is a *successor* of  $X$ . The algorithm to solve problem *Predicate\_Rel* is given in two parts. The processing on each of the  $n$  processes  $P_1$  to  $P_n$  is given next. The processing by  $P_0$  is given in Section 3.

### Processing at $P_i$ , ( $1 \leq i \leq n$ )

Each process  $P_i$ , where  $1 \leq i \leq n$ , maintains the data structure  $Log_i$  that contains the information of the start and end of the (latest) interval to be sent to  $P_0$ .  $Log_i$  is constructed and sent to  $P_0$  using the protocol shown in Figure 4.  $P_0$  uses the *Logs* reported to determine the relationship between interval pairs.

### Complexity Analysis at $P_i$ ( $1 \leq i \leq n$ )

**Space Complexity of  $Log$ .** at each  $P_i$ ,  $1 \leq i \leq n$ . Each  $Log$  at a process stores the start ( $t^-$ ) and the end ( $t^+$ ) of an interval. As only one  $Log$  entry exists at a time, the space needed at a process  $P_i$  at any time is 2 integers.

**Space Complexity of Control Messages.** sent to  $P_0$  by processes  $P_1$  to  $P_n$ .

- As one message is sent per interval, the number of messages is  $p$  for each  $P_i$  ( $i \neq 0$ ). This gives a total number of messages as  $np$ .
- The size of each message is 2 as each message contains a  $Log$ . The total message space overhead for any process is the sum over all the  $Logs$  for that process, which is  $2p$ . Hence the total message space complexity is  $2np$ .

## 3 Algorithm *Predicate\_Rel*

The algorithm detects a set of intervals, one on each process, such that each pair of intervals satisfies the relationship specified for that pair of processes. If no such set of intervals exists, the algorithm does not return any interval set. The central process  $P_0$  maintains  $n$  queues, one for *Logs* from each process and determines which relation holds between pairs of intervals. The queues are processed using the formalism in [3, 4, 5]. If there exists an interval at the head of each queue and these intervals cannot be pruned, then these intervals satisfy  $r_{i,j} \forall i, j$ , where  $i \neq j$  and  $1 \leq i, j \leq n$ . Hence these intervals form a solution set.

We use the *prohibition* function  $\mathcal{H}(r_{i,j})$  and the *allows* relation  $\rightsquigarrow$  [3, 4, 5, 6]. For each  $r_{i,j} \in \mathfrak{R}$ , its *prohibition function*  $\mathcal{H}(r_{i,j})$  is the set of all relations  $R$  such

**Table 2.** Prohibition functions  $\mathcal{H}(r_{i,j})$  for the 13 independent relations  $r_{i,j}$  in  $\mathfrak{R}$

Relation $r$	$\mathcal{H}(r_{i,j}(X_i, Y_j))$	$\mathcal{H}(r_{j,i}(Y_j, X_i))$
$p = pb^{-1}$	$\emptyset$	$\{p, m, mb, o, ob, s, sb, f, fb, c, cb, q\}$
$m = mb^{-1}$	$\{p, m, o, s, f, fb, cb, q\}$	$\{p, m, mb, o, ob, s, sb, f, fb, c, cb, q\}$
$o = ob^{-1}$	$\{p, m, o, s, f, fb, cb, q\}$	$\{p, m, mb, o, ob, s, sb, f, fb, c, cb, q\}$
$s = sb^{-1}$	$\{p, m, mb, o, ob, s, sb, f, fb, c, cb, q\}$	$\{p, m, mb, o, ob, s, sb, f, fb, c, cb, q\}$
$f = fb^{-1}$	$\{p, m, o, s, f, fb, cb, q\}$	$\{p, m, mb, o, ob, s, sb, f, fb, c, cb, q\}$
$c = cb^{-1}$	$\{p, m, o, s, f, fb, cb, q\}$	$\{p, m, mb, o, ob, s, sb, f, fb, c, cb, q\}$
$q = q^{-1}$	$\{p, m, mb, o, ob, s, sb, f, fb, c, cb, q\}$	$\{p, m, mb, o, ob, s, sb, f, fb, c, cb, q\}$

that if  $R(X, Y)$  is true, then  $r_{i,j}(X, Y')$  can never be true for some successor  $Y'$  of  $Y$ .  $\mathcal{H}(r_{i,j})$  is the set of relations that prohibit  $r_{i,j}$  from being true in the future. Two relations  $R'$  and  $R''$  in  $\mathfrak{R}$  are related by the *allows* relation  $\rightsquigarrow$  if the occurrence of  $R'(X, Y)$  does not prohibit  $R''(X, Y')$  for some successor  $Y'$  of  $Y$ .

**Definition 1.** Function  $\mathcal{H} : \mathfrak{R} \rightarrow 2^{\mathfrak{R}}$  is defined to be  $\mathcal{H}(r_{i,j}) = \{R \in \mathfrak{R} \mid \text{if } R(X, Y) \text{ is true then } r_{i,j}(X, Y') \text{ is false for all } Y' \text{ that succeed } Y\}$ .

**Definition 2.**  $\rightsquigarrow$  is a relation on  $\mathfrak{R} \times \mathfrak{R}$  such that  $R' \rightsquigarrow R''$  if the following holds. If  $R'(X, Y)$  is true then  $R''(X, Y')$  can be true for some  $Y'$  that succeeds  $Y$ .

**Examples**

- (i)  $c \rightsquigarrow o$  because if  $c(X, Y)$  is true, then there is a possibility that  $o(X, Y')$  is also true, where  $Y'$  succeeds  $Y$ .
- (ii)  $m^{-1} \rightsquigarrow f$  because if  $m^{-1}(X, Y)$  is true, then there is a possibility that  $f(X, Y')$  is also true, where  $Y'$  succeeds  $Y$ .

**Lemma 1.** If  $R \in \mathcal{H}(r_{i,j})$  then  $R \not\rightsquigarrow r_{i,j}$  else if  $R \notin \mathcal{H}(r_{i,j})$  then  $R \rightsquigarrow r_{i,j}$ .

**Proof.** If  $R \in \mathcal{H}(r_{i,j})$ , using Definition 1, it can be inferred that  $r_{i,j}$  is false for all  $Y'$  that succeed  $Y$ . This does not satisfy Definition 2. Hence  $R \not\rightsquigarrow r_{i,j}$ . If  $R \notin \mathcal{H}(r_{i,j})$ , it follows that  $r_{i,j}$  can be true for some  $Y'$  that succeeds  $Y$ . This satisfies Definition 2 and hence  $R \rightsquigarrow r_{i,j}$ . □

Table 2 gives  $\mathcal{H}(r_{i,j})$  for the 13 interaction types in  $\mathfrak{R}$ . It is constructed by analyzing each interaction pair.

**Example:** The third row of Table 2 gives the relations  $o$  and  $ob$ .

- In column two,  $\mathcal{H}(o_{i,j}(X_i, Y_j)) = \{p, m, o, s, f, fb, cb, q\}$ . Hence,  $p(X_i, Y_j)$  or  $m(X_i, Y_j)$  or  $o(X_i, Y_j)$  or  $s(X_i, Y_j)$  or  $f(X_i, Y_j)$  or  $fb(X_i, Y_j)$  or  $cb(X_i, Y_j)$  or  $q(X_i, Y_j)$  implies that  $o(X_i, Y'_j)$  can never hold for any successor  $Y'_j$  of  $Y_j$ .
- In column three,  $\mathcal{H}(ob_{j,i}(Y_j, X_i)) = \{p, m, mb, o, ob, s, sb, f, fb, c, cb, q\}$ . Hence,  $p(Y_j, X_i)$  or  $m(Y_j, X_i)$  or  $mb(Y_j, X_i)$  or  $o(Y_j, X_i)$  or  $ob(Y_j, X_i)$  or  $s(Y_j, X_i)$  or  $sb(Y_j, X_i)$  or  $f(Y_j, X_i)$  or  $fb(Y_j, X_i)$  or  $c(Y_j, X_i)$  or  $cb(Y_j, X_i)$  or  $q(Y_j, X_i)$  implies that  $ob(Y_j, X'_i)$  can never hold for any successor  $X'_i$  of  $X_i$ .

The following theorem states that if  $R'$  allows  $R''$ , then Theorem 1 states that  $R'^{-1}$  necessarily does not allow relation  $R''^{-1}$ .

**Table 3.** The “allows” relation  $\rightsquigarrow$  on  $\mathfrak{R} \times \mathfrak{R}$ , in matrix form, to verify Theorem 1

$\rightsquigarrow$	$p$	$pb$	$m$	$mb$	$o$	$ob$	$s$	$sb$	$f$	$fb$	$c$	$cb$	$q$
$p$	1												
$pb$	1	1	1	1	1	1	1	1	1	1	1	1	1
$m$	1												
$mb$	1		1		1			1		1			
$o$	1												
$ob$	1		1		1			1		1			
$s$	1												
$sb$	1		1		1			1		1			
$f$	1												
$fb$	1												
$c$	1		1		1			1		1			
$cb$	1												
$q$	1												

**Theorem 1.** For  $R', R'' \in \mathfrak{R}$  and  $R' \neq R''$ , if  $R' \rightsquigarrow R''$  then  $R'^{-1} \not\rightsquigarrow R''^{-1}$ .

The theorem can be observed to be true from Lemma 1 and Table 2 by using a case-by-case analysis. Table 3 shows the grid of the  $\rightsquigarrow$  relation for this analysis. A “1” indicates that the row header allows the column header. Alternately, this analysis is easier by using the following form of Theorem 1: “For  $R' \neq R''$ , if  $R' \notin \mathcal{H}(R'')$ , then  $R'^{-1} \in \mathcal{H}(R''^{-1})$ ”.

(Example 1.)  $c \rightsquigarrow o \Rightarrow c^{-1} \not\rightsquigarrow o^{-1}$ , which is true.  
 (Example 2:)  $m^{-1} \rightsquigarrow f \Rightarrow m \not\rightsquigarrow f^{-1}$ , which is true.

Note  $R' \neq R''$  in Theorem 1; otherwise  $R' \rightsquigarrow R'$  holds as for  $p$ ,  $pb$ , and  $c$ , leading to  $R'^{-1} \not\rightsquigarrow R'^{-1}$ , a contradiction.

**Lemma 2.** If the relationship  $R(X, Y)$  between intervals  $X$  at  $P_i$  and  $Y$  at  $P_j$  is contained in the set  $\mathcal{H}(r_{i,j})$  and  $R \neq r_{i,j}$ , then  $X$  can be removed from the queue  $Q_i$ .

**Proof.** By definition of  $\mathcal{H}(r_{i,j})$ ,  $r_{i,j}(X, Y')$  cannot exist, where  $Y'$  is any successor of  $Y$ . As  $r_{i,j} \neq R$ ,  $X$  cannot be a part of the solution. So  $X$  can be deleted. □

**Lemma 3.** If the relationship between a pair of intervals  $X$  at  $P_i$  and  $Y$  at  $P_j$  is not equal to  $r_{i,j}$ , then either  $X$  or  $Y$  is removed from the queue.

**Proof.** We use contradiction. Assume relation  $R(X, Y)$  ( $\neq r_{i,j}(X, Y)$ ) is true for intervals  $X$  and  $Y$ . From Lemma 2, the only time neither  $X$  nor  $Y$  will be deleted is when  $R \notin \mathcal{H}(r_{i,j})$  and  $R^{-1} \notin \mathcal{H}(r_{j,i})$ . From Lemma 1, it can be inferred that  $R \rightsquigarrow r_{i,j}$  and  $R^{-1} \rightsquigarrow r_{j,i}$ . As  $r_{i,j}^{-1} = r_{j,i}$ , we get  $R \rightsquigarrow r_{i,j}$  and  $R^{-1} \rightsquigarrow r_{i,j}^{-1}$ . This is a contradiction as by Theorem 1,  $R$  being unequal to  $r_{i,j}$ ,  $R \rightsquigarrow r_{i,j} \Rightarrow$

$R^{-1} \not\rightsquigarrow r_{i,j}^{-1}$ . Hence  $R \in \mathcal{H}(r_{i,j})$  or  $R^{-1} \in \mathcal{H}(r_{j,i})$  or both; so one or both of  $X$  and  $Y$  can be deleted.  $\square$

Lemma 3 guarantees progress; when two intervals are checked, if the desired relationship is not satisfied, at least one of them can be discarded. Further, it is possible that both the intervals being tested are discarded.

**Example:** We want to detect  $X$  and  $Y$ , where  $r_{i,j}(X, Y) = f$ . If  $R(X, Y) = o$ , we have that  $o \not\rightsquigarrow f$ ; hence  $o(X, Y)$  will not allow  $f(X, Y')$  to be true for any  $Y'$ . Hence  $X$  must be deleted. Further,  $ob \not\rightsquigarrow fb$  and hence  $ob(Y, X)$  will not allow  $fb(Y, X')$  to be true for any  $X'$ . Hence,  $Y$  must also be deleted.

**Theorem 2.** *Problem Predicate\_Rel is solved by the algorithm in Figure 5.*

**Proof.** The algorithm implements Lemma 2 which allows queues to be pruned correctly. An interval gets deleted only if it cannot be part of the solution. Specifically, interval  $X$  gets deleted if  $R(X, Y) \in \mathcal{H}(r_{i,j})$  and  $R \neq r_{i,j}$  (lines 13,14,17). Similarly,  $Y$  is deleted if  $R(Y, X) \in \mathcal{H}(r_{j,i})$  and  $R \neq r_{j,i}$  (lines 15-17). Further, each interval gets examined unless a solution is found using one of its predecessors. Lemma 3 guarantees that if  $R(X, Y) \neq r_{i,j}$ , then either interval  $X$  or interval  $Y$  is deleted. Hence, if every queue is non-empty and its head cannot be pruned, then the set of intervals at the head of each queue forms a solution.

The set *updatedQs* stores the indices of all the queues whose heads get updated. In each iteration of the **while** loop, the indices of all the queues whose heads satisfy Lemma 2 are stored in set *newUpdatedQs* (lines (13)-(16)). In lines (17) and (18), the heads of all these queues are deleted and indices of the updated queues are stored in the set *updatedQs*. Observe that only interval pairs which were not compared earlier are compared in subsequent iterations of the **while** loop. The loop runs until no more queues can be updated. If all the queues are now non-empty, then a solution is found (Lemma 3), where for the intervals  $X = head(Q_i)$  and  $Y = head(Q_j)$ ,  $R(X, Y) = r_{i,j}$ .  $\square$

**Theorem 3.** *The algorithm in Figure 5 has the following complexities.*

1. *The total message space complexity is  $2np$ . (proved in Section 2)*
2. *The total space complexity at process  $P_0$  is  $2np$ . (follows from (1))*
3. *The time complexity at  $P_0$  is  $O((n - 1)pn)$ .*

**Proof.** The time complexity is the product of the number of steps needed to determine a relationship ( $O(1)$ , follows trivially from Figure 2) and the number of relations determined. For each interval considered from one of the queues in *updatedQs* (lines (6)-(12)), the number of relations determined is  $n - 1$ . Thus the number of relations determined for each iteration of the **while** loop is  $(n - 1)|updatedQs|$ . But  $\sum |updatedQs|$  over all iterations of the **while** loop is less than the total number of intervals over all the queues. Thus, the total number of relations determined is less than  $(n - 1) \cdot x$ , where  $x = pn$  is the upper bound on the total number of intervals over all the queues. As the time required to determine a relationship is  $O(1)$ , the time complexity is  $O((n - 1)np)$ .  $\square$

**queue of Log:**  $Q_1, Q_2, \dots, Q_n = \perp$   
**set of int:**  $updatedQs, newUpdatedQs = \{\}$   
On receiving interval from process  $P_z$  at  $P_0$   
 1: Enqueue the interval onto queue  $Q_z$   
 2: **if** (number of intervals on  $Q_z$  is 1) **then**  
 3:      $updatedQs = \{z\}$   
 4:     **while** ( $updatedQs$  is not empty)  
 5:          $newUpdatedQs = \{\}$   
 6:         **for** each  $i \in updatedQs$   
 7:             **if** ( $Q_i$  is non-empty) **then**  
 8:                  $X = \text{head of } Q_i$   
 9:                 **for**  $j = 1$  to  $n$   
 10:                     **if** ( $Q_j$  is non-empty) **then**  
 11:                          $Y = \text{head of } Q_j$   
 12:                         Test for  $R(X, Y)$  using interval timestamps (Fig. 2)  
 13:                         **if** ( $R(X, Y) \in \mathcal{H}(r_{i,j})$ ) and  $R \neq r_{i,j}$  **then**  
 14:                              $newUpdatedQs = \{i\} \cup newUpdatedQs$   
 15:                         **if** ( $R(Y, X) \in \mathcal{H}(r_{j,i})$ ) and  $R \neq r_{j,i}$  **then**  
 16:                              $newUpdatedQs = \{j\} \cup newUpdatedQs$   
 17:                         Delete heads of all  $Q_k$  where  $k \in newUpdatedQs$   
 18:                          $updatedQs = newUpdatedQs$   
 19:             **if** (all queues are non-empty) **then**  
 20:                 Heads of queues identify intervals that form the solution.

**Fig. 5.** On-line algorithm at  $P_0$  to solve *Predicate\_Rel*, based on [4, 5]

## 4 Conclusions and Discussion

This paper formulated the problem of detecting a global predicate in a (distributed) ubiquitous system assuming the presence of global time. Such ubiquitous systems are becoming common due to the spread of embedded devices that are networked together, sensor networks, and ad-hoc networks. The assumption of an approximate global time axis is also becoming reasonable in an increasing number of scenarios due to the spread and availability of GPS, and inexpensive clock synchronization algorithms. The paper presented an algorithm based on [4, 5] to detect a global predicate specified across the various locations, assuming that event streaming from those locations to a central location is available. This model is reasonable because the dynamic ambient network in the ubiquitous environment connects each device to more powerful processing resources. The proposed algorithm is highly scalable as it has low overhead.

We mention some limitations of the approach. Global time is at best an approximation. A common time axis will not be applicable when predicates based on causality (i.e., happens before) relation [14] are specified. In such cases, the algorithms presented in [3, 4] using the theory in [11, 12] can be used. Also, the availability of global time in some scenarios with limited resources and/or constrained network topologies may not be practical.

The presented formalism assumed *local discreteness* which implied *local interval separation* and no *points*. Variations can be handled by adapting this formalism

(see [13]). For example, if *local interval separation* is relaxed, an interval can begin at the same instant the previous interval at the same process ends.  $X'_i$  would be a successor of  $X_i$  if  $m(X_i, X'_i)$  or  $p(X_i, X'_i)$ . In Table 2,  $\mathcal{H}(m)$  would exclude  $f, fb, q$ , and  $\mathcal{H}(s), \mathcal{H}(sb), \mathcal{H}(q)$  would each exclude  $mb$ . In Table 3 for  $\sim$ , there would be “1” for  $(f, m), (fb, m), (q, m), (mb, s), (mb, sb), (mb, q)$ . Theorem 1 can be seen to still hold. Other variations, such as allowing *points* (one point per clock tick), and about clock properties and time density, can be similarly handled.

## References

1. I. Akyildiz, W. Su, Y. Sankarasubramanian, E. Cayirci, Wireless sensor networks: A survey, *Computer Networks*, 38(4): 393-422, 2002.
2. J. Allen, Maintaining knowledge about temporal intervals, *Communications of the ACM*, 26(11): 832-843, 1983.
3. P. Chandra, A. D. Kshemkalyani, Detection of orthogonal interval relations, *Proc. 9th High-Performance Computing Conference (HiPC)*, LNCS 2552, Springer, 323-333, 2002.
4. P. Chandra, A. D. Kshemkalyani, Global predicate detection under fine-grained modalities, *Proc. ASIAN Computing Conference 2003 (ASIAN)*, LNCS 2896, Springer, 91-109, Dec. 2003.
5. P. Chandra, A. D. Kshemkalyani, Causality-based predicate detection across space and time, *IEEE Transactions on Computers*, 54(11): 1438-1453, 2005.
6. P. Chandra, A. D. Kshemkalyani, Global state detection based on peer-to-peer interactions, *Proc. IFIP Conf. on Embedded and Ubiquitous Computing (EUC)*, LNCS, Springer, Dec. 2005.
7. K. M. Chandy, L. Lamport, Distributed snapshots: Determining global states of distributed systems, *ACM Trans. Computer Systems*, 3(1): 63-75, 1985.
8. J. Elson, K. Romer, Wireless sensor networks: A new regime for time synchronization, *First Workshop on Hot Topics In Networks (HotNets-I)*, October 2002.
9. S. Ganeriwal, R. Kumar, M. Srivastava, Timing-sync protocol for sensor networks, *Proc. ACM Conf. Embedded Networked Sensor Systems*, 138-149, Nov. 2003.
10. C. L. Hamblin, Instants and intervals, in “The Study of Time,” pp. 324-332, Springer-Verlag New York/Berlin, 1972.
11. A. D. Kshemkalyani, Temporal interactions of intervals in distributed systems, *Journal of Computer and System Sciences*, 52(2): 287-298, April 1996.
12. A. D. Kshemkalyani, A fine-grained modality classification for global predicates, *IEEE Trans. Parallel and Distributed Systems*, 14(8): 807-816, August 2003.
13. A. D. Kshemkalyani, Predicate detection using event streams in ubiquitous environments, *UIC Technical Report UIC-CS-02-05*, 2005.
14. L. Lamport, Time, clocks, and the ordering of events in a distributed system, *Communications of the ACM*, 21(7): 558-565, July 1978.
15. T. Logsdon, *The Navstar Global Positioning System*, Van Nostrand/Reinhold, New York, 1992.
16. D. Mills, Internet time synchronization: the Network Time Protocol, *IEEE Trans. on Communications*, 39(10): 1482-1493, October 1991.
17. K. Romer, Time synchronization in ad-hoc networks, *Proc. ACM MobiHoc*, 2001.
18. B. Sundararaman, U. Buy, A. D. Kshemkalyani, Clock synchronization for wireless sensor networks: A survey, *Ad-Hoc Networks*, 3(3): 281-323, May 2005.
19. S. Tilak, N. Abu-Ghazaleh, W. Heinzelman, A taxonomy of wireless micro-sensor models, *ACM Mobile Computing & Communications Review*, 6(2), April 2002.



# Image Watermarking Technique Based on Two-Dimensional Chaotic Stream Encryption

Hanping Hu<sup>1</sup> and Yongqiang Chen<sup>1,2</sup>

<sup>1</sup> Institute for Pattern Recognition and Artificial Intelligence,  
State Education Department Key Laboratory for Image Processing and Intelligent Control,  
Huazhong University of Science and Technology, Wuhan 430074, China

<sup>2</sup> Department of Computer and Information Engineering,  
Wuhan Polytechnic University, Wuhan 430023, China  
chenyqwh@163.com

**Abstract.** This paper proposes a kind of wavelet domain image digital watermarking technique using two-dimensional chaotic stream encryption and human visual model. A stream encryption algorithm based on two-dimensional Logistic chaotic map is researched and realized for meaningful grayscale watermarking image. The block embedding intensity is calculated and combined with the human visual model, so that the embedding and detection steps of encrypted binary watermark can be adaptively fulfilled in the wavelet coefficients of the host image. The experimental results have shown that this watermarking technique can endure regular digital image processing and have preferable performance.

## 1 Introduction

Digital watermark is a kind of technology that embeds copyright information into multimedia data. Unlike encryption, the watermark remains in the content in its original form and does not prevent a user from listening to, viewing, examining, or manipulating the content. Digital watermark technology opens a new door to authors, producers, publishers, and service providers for protecting their rights and interests in multimedia documents<sup>[1]</sup>.

Considering the watermark security and embedded information's secrecy, we can use the cipher technology to encrypt the information codes and positions. The chaotic encryption technology is a novel method that has good performance and has been developed in recent years. Currently, most image watermarking schemes usually embed chaotic signals produced by given chaotic system into host image<sup>[2]</sup>. Methods that encrypt meaningful watermarking signals are mainly based on one-dimensional chaotic system, but methods based on two-dimension are rarely found<sup>[3]</sup>. Besides, attack methods aiming at low dimensional chaotic systems have been found, so the security couldn't be guaranteed.

This paper employs the two-dimensional chaotic Logistic map to encrypt the meaningful gray image<sup>[4,5]</sup>. Based on HVS model<sup>[6-8]</sup>, we embed the encrypted binary image into the wavelet domain of host image to get more security.

## 2 Two-Dimensional Logistic Map

As the encryption using chaotic sequence produced by one-dimensional Logistic system is weak in security, we have to turn to two-dimensional Logistic system.

### 2.1 The Definition of Two-Dimensional Logistic Map

The two-dimensional Logistic map can be defined as:

$$\begin{cases} x_{n+1} = 4\mu_1 x_n(1 - x_n) + g_1(x_n, y_n) \\ y_{n+1} = 4\mu_2 y_n(1 - y_n) + g_2(x_n, y_n) \end{cases} \tag{1}$$

$g_1$  and  $g_2$  are coupled terms, which can be used in two modes:

- (1)  $g_1 = \gamma y_n$  and  $g_2 = \gamma x_n$ , and both are simple coupled terms;
- (2)  $g_1 = g_2 = \gamma x_n y_n$ , and both are symmetry quadratic coupled terms.

Adopting the mode of simple coupled term, we change equation (1) as follows:

$$\begin{cases} x_{n+1} = 4\mu_1 x_n(1 - x_n) + \gamma y_n \\ y_{n+1} = 4\mu_2 y_n(1 - y_n) + \gamma x_n \end{cases} \tag{2}$$

The dynamical behavior of this system is controlled by control parameters of  $\mu_1$ ,  $\mu_2$  and  $\gamma$ . These parameters must be suitably chosen to control the system.

### 2.2 Chaos of the Two-Dimensional Logistic Map

Nonlinear dynamical system will evolve into the ultimate set, namely the attractor, whose dimension is less than the phase space's. With the control parameters changing, the simple attractor will develop into the strange attractor and the system becomes chaotic. Investigating into the chaotic movement, we will observe the bifurcation figure or phase figure. However, due to chaotic complexity, some chaotic criterions are needed, such as reconstruction of phase space, power spectrum analysis, information dimension, Lyapunov exponent and metric entropy. The Lyapunov exponent calculated by difference equation group is a statistical eigenvalue depicting chaotic movement and a measurement unit of average constringency or radiation of neighboring orbits in phase space.

For the two-dimensional Logistic map, its Jacobi matrix  $f'(z)$  is represented as:

$$f'(z) = \frac{\partial f}{\partial z} = \begin{bmatrix} 4\mu_1 - 8\mu_1 x & \gamma \\ 4\mu_2 - 8\mu_2 y & \gamma \end{bmatrix} \tag{3}$$

Having  $J_i = f'(z_0) \cdot f'(z_1) \cdots f'(z_{i-1}) = [f^i(z_{i-1})]_{z=z_0}^1$ , we can compute the module of the 2 complex latent roots of  $J_i$  and permute them as  $|\lambda_1^{(i)}| \geq |\lambda_2^{(i)}|$ , where the

Lyapunov exponent is  $\lambda_k = \lim_{i \rightarrow \infty} \frac{1}{i} \ln |\lambda_k^{(i)}|$ ,  $k=1,2$ . If  $\lambda_1$  is positive, the system would be chaotic.

When  $\mu_1 = \mu_2 = \mu \in [0.6, 0.9]$  and  $\gamma = 0.1$ , we can analyze the Lyapunov exponent figure and bifurcation figure. The system moves periodically where  $\mu < 0.815$  and  $\lambda_1 < 0$ . The system is chaotic where  $0.815 < \mu < 0.89$  and mostly  $\lambda_1 > 0$ , but the system locates in periodic window with different periods in chaotic area where  $\lambda_1 < 0$  in some narrow regions. The system is chaotic where  $\mu \geq 0.89$  and  $\lambda_1 > 0$ . According to the above analysis, the map system, when satisfying the chaotic movement conditions, can be used in digital image encryption.

### 3 Digital Image Stream Encryption Algorithm

Stream cipher is a kind of symmetric algorithm, which encrypts one bit in the plaintext once a time, and can be regarded as a block cipher of which the block length is one. In the condition of fault transmission, there isn't false spread for stream cipher.

Let  $\mathbf{F} = [F_{s,t}]_{M \times M}$  represent a gray watermarking image having  $L$  gray levels with size  $M \times M$  ( $1 \leq s, t \leq M$ ).  $F_{s,t}$  ( $0 \leq F_{s,t} \leq L-1$ ) is the decimal gray value of the pixel  $(s, t)$ .  $x_p$  and  $y_p$  are values obtained after the map system is iterated  $P$  times. The encryption algorithm is described as follows:

(1) Transform  $F_{s,t}$  into the binary sequence  $m_{s,t} = m_{s,t,1}m_{s,t,2} \cdots m_{s,t,l}$ ,  $l = \lceil \log_2 L \rceil$ .

The watermarking image will be represented by a binary matrix  $\mathbf{m} = [m_{s,t}]$  with  $M$  rows and  $Ml$  columns.

(2) Transform the decimal fraction of  $x_i$  into binary sequence and choose the first  $l$  bits to be represented as  $x_{i,1}x_{i,2} \cdots x_{i,l}$ . Like  $x_i$ , the decimal fraction of  $y_i$  is represented as  $y_{i,1}y_{i,2} \cdots y_{i,l}$ .

(3) According to the row order, do the XOR operation  $c_{s,t,j}^1 = m_{s,t,j} \oplus x_{i,j}$ ,  $P \leq i \leq P+M^2$ ,  $1 \leq j \leq l$ , so the binary sequences of image are encrypted firstly and  $c_{s,t}^1 = c_{s,t,1}^1 c_{s,t,2}^1 \cdots c_{s,t,l}^1$  is got.

(4) According to the column order, do the XOR operation  $c_{s,t,j}^2 = c_{s,t,j}^1 \oplus y_{i,j}$  and get  $c_{s,t}^2 = c_{s,t,1}^2 c_{s,t,2}^2 \cdots c_{s,t,l}^2$ .

(5) Revert the binary sequence  $c_{s,t}^2$  into the decimal value represented as  $W_{s,t}$  ( $0 \leq W_{s,t} \leq L-1$ ) and get the encrypted image  $\mathbf{W} = [W_{s,t}]_{M \times M}$ .

Let  $\widehat{\mathbf{W}} = [\widehat{W}_{s,t}]_{M \times M}$  represent the received encrypted image and the pixel gray value is  $\widehat{W}_{s,t}$ . The decryption procedure is described as follows:

- (1) Transform the gray value  $\widehat{W}_{s,t}$  into the binary sequence  $\widehat{c}_{s,t}^2$ .
- (2) The decimal fraction of  $x_i$  and  $y_i$  are denoted by  $x_{i,1}x_{i,2}\cdots x_{i,l}$  and  $y_{i,1}y_{i,2}\cdots y_{i,l}$  respectively.
- (3) Do XOR operation  $\widehat{c}_{s,t,j}^1 = \widehat{c}_{s,t,j}^2 \oplus y_{i,j}$ , and decrypt the gray value according to the column order and get  $\widehat{c}_{s,t}^1$ .
- (4) Do XOR operation  $\widehat{m}_{s,t,j}^1 = \widehat{c}_{s,t,j}^1 \oplus x_{i,j}$ , and decrypt the gray value according to the row order and get  $\widehat{m}_{s,t}$ . Transform binary value  $\widehat{m}_{s,t}$  into decimal value  $\widehat{F}_{s,t}$  and get decrypted image  $\widehat{\mathbf{F}} = [\widehat{F}_{s,t}]_{M \times M}$ .

From the above encryption and decryption algorithms, it can be concluded that if the received image  $\widehat{W}(s,t)$  has not been processed, the equations  $\widehat{\mathbf{W}} = \mathbf{W}$  and  $\widehat{\mathbf{F}} = \mathbf{F}$  are satisfied after the decryption.

In the watermark embedding step, we will use the binary form of the encrypted image, namely  $\mathbf{W} = [W_{i,j}]_{M \times Ml}$ ,  $W_{i,j} = 0,1 (1 \leq i \leq M, 1 \leq j \leq Ml)$ .

The key  $k(\mu_1, \mu_2, \gamma, x_0, y_0, P)$  is made up of the parameters and the initial value of chaotic system. Because the chaotic system is very sensitive to the parameters and initial value, the theoretic key space is infinite. Therefore, we can almost assign one unique key for one encryption process. The encryption algorithm accords with the Kerckhoff's rules and is a modern encryption method.

In order to demonstrate the algorithm's validity and security, a gray image of face with 256 gray levels and the size of  $64 \times 64$ , namely  $M = 64, L = 256, l = 8$ , is selected from ORL face database<sup>[9]</sup>. Assume control parameters  $\mu_1 = \mu_2 = 0.9, \gamma = 0.1$  and initial values  $x_0 = 0.1, y_0 = 0.11, P = 500$ . The encryption and decryption results are described in Fig.1, in which (a) is the original image, (b) is the result of the original image encrypted only in the rows, (c) is the result of image encrypted in the rows and columns, (d) is the result of the decrypted image with right parameters. From Fig.1, we can observe that if the face image is only encrypted in the row, the sketch of face could still be found; but through encryption in both rows and columns, the resultant image is fully disordered. If only changed one parameter in decrypting step, the result would be false. From the above experiments, it can be concluded that this algorithm has simple complexity and good encrypting effect.

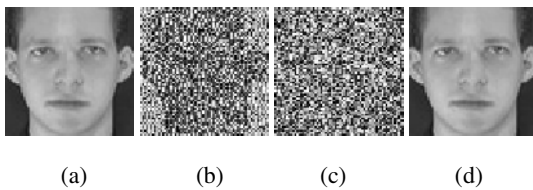


Fig. 1. Encrypted and decrypted image

## 4 Image Watermarking Algorithm

According to the embedding technology, the digital image watermark can be classified into two kinds: spatial watermark and frequency domain watermark. Because the frequency domain watermark is more robust than the spatial watermark, the adaptive watermarking algorithm in DWT domain of host image is adopted based on the HVS model.

### 4.1 HVS Model

According to the HVS model<sup>[8]</sup>, the frequency response function is shown as:

$$H(\omega) = (a + b\omega)\exp(-c\omega) \quad (4)$$

$\omega$  is the radial frequency,  $a, b$  and  $c$  are parameters determining the HVS curve shape. When  $\omega_{\max} = 3$ , the HVS curve shape can be expressed as:

$$H(\omega) = (0.2 + 0.45\omega)\exp(-0.18\omega) \quad (5)$$

The image coding using DFT is equivalent to do symmetry spread to original image, but the human eye can't observe this phenomenon. Therefore, the HVS needs a correctional function  $A(\omega)$  and the frequency function can be rewritten as:

$$\widehat{H}(\omega) = H(\omega)|A(\omega)| = \begin{cases} 0.05 \exp(\omega^{0.554}) & \omega < 7 \\ \exp(-9|\lg \omega - \lg 9|^{2.3}) & \omega \geq 7 \end{cases} \quad (6)$$

$\omega = \omega_d \omega_s$ ,  $\omega_d = (u^2 + v^2)^{0.5} / 2N$ ,  $u, v = 0, \dots, N-1$ .  $N$  is the size of the DFT block.  $\omega_s$  is a sampling function of the observation distance (assume  $\omega_s = 48$ ). The corresponding  $\widehat{H}(\omega)$  of every DFT coefficient  $(u, v)$  can be computed. Thereby, the function  $\widetilde{H}(u, v)$ , corresponding with conjugated  $(u, v)$ , is obtained.

### 4.2 Watermark Embedding and Extracting

Wavelet transform accords with some characters of the HVS and strengthens the imperceptibility of watermark. Colligated the HVS model and wavelet transform technology, the encrypted watermarking image is adaptively embedded into the middle frequency subbands of the host image based on the value of the frequency response function.

Let  $\mathbf{I} = [I_{i,j}]_{N \times N}$  ( $1 \leq i, j \leq N$ ) represent the host image with size  $N \times N$  and  $\widehat{\mathbf{I}} = [\widehat{I}_{i,j}]_{N \times N}$  denotes the watermarked image. The detailed steps of the embedding algorithm are described as follows:

(1) Use DWT to transform the host image  $\mathbf{I}$  and get middle frequency subbands  $f^{(str)} = [f_{i,j}^{(str)}]_{\frac{N}{2} \times \frac{N}{2}}$  ( $1 \leq i, j \leq \frac{N}{2}$ ),  $str \in \{\text{HL}, \text{LH}\}$ .

(2) Divide the binary encrypted watermark  $\mathbf{W} = [W_{i,j}]_{M \times Ml}$  ( $1 \leq i \leq M, 1 \leq j \leq Ml$ ) into two no-overlap blocks  $\mathbf{W}^{(str)} = [W_{i,j}^{(str)}]_{\frac{N}{2} \times \frac{N}{2}}$  ( $1 \leq i, j \leq \frac{N}{2}$ ).

(3) According to the size  $(\frac{N}{4}) \times (\frac{N}{4})$ , divide the middle frequency subband into 4 no-overlap blocks  $f_k^{(str)}(x, y), k = 1, 2, 3, 4$ .

(4) According to the block position, compute the watermark embedding intensity  $\alpha^{(str)} = \frac{1}{4} \sum_{k=1}^4 \alpha_k^{(str)}, str \in \{HL, LH\}$ , where  $\alpha_k^{(str)}$  is defined as:

$$\alpha_k^{(str)} = \beta \left[ \sum \tilde{H}(u, v) \left| \hat{f}_k^{(str)}(u, v) \right|^2 \right]^{\frac{1}{2}} / \max_k \left[ \sum \tilde{H}(u, v) \left| \hat{f}_k^{(str)}(u, v) \right|^2 \right]^{\frac{1}{2}}$$

$\left| \hat{f}_k^{(str)}(u, v) \right|$ , coming from  $f_k^{(str)}(x, y)$ , can be calculated by transforming the image using DFT.  $\beta$  equals to one tenth of the mean gray value of the encrypted watermark.

(5) Use the equation  $\hat{f}_{i,j}^{(str)} = f_{i,j}^{(str)} + \alpha^{(str)} W_{i,j}^{(str)}$  to revise the DWT coefficients of host image, then do IDWT and get watermarked image  $\hat{\mathbf{I}} = [\hat{I}_{i,j}]_{N \times N}$ .

The watermark detection is to detect whether the embedded watermark information exists in the prepared detection image  $\tilde{\mathbf{I}} = [\tilde{I}_{i,j}]_{N \times N}$  ( $1 \leq i, j \leq N$ ). The host image or watermark image is needful to determine whether watermark exists in the  $\tilde{\mathbf{I}}$  through computing the correlation coefficient between  $\mathbf{W}$  with  $\tilde{\mathbf{W}}$  extracted from  $\tilde{\mathbf{I}}$ . The watermark extracting steps are described as follows:

(1) Same as the 1<sup>st</sup>, 3<sup>rd</sup>, 4<sup>th</sup> embedding steps, do the DWT, divide subband into blocks and compute  $\alpha^{(str)}$ .

(2) Use the equation  $\tilde{W}_{i,j}^{(str)} = (\tilde{f}_{i,j}^{(str)} - f_{i,j}^{(str)}) / \alpha^{(str)}$  to get sub-block  $\tilde{\mathbf{W}}^{(str)}$ , and then unit the sub-blocks into  $\tilde{\mathbf{W}}$ .

Set the threshold value  $T$  and use the similarity equation  $sim(\mathbf{W}, \tilde{\mathbf{W}}) = (\mathbf{W}\tilde{\mathbf{W}}) / \sqrt{\mathbf{W}\mathbf{W}}$ . If  $sim(\mathbf{W}, \tilde{\mathbf{W}}) \geq T$ , we can determine that the watermark exist in the image  $\tilde{\mathbf{I}}$ . The above-mentioned decryption algorithm is used to decrypt  $\tilde{\mathbf{W}}$  and examine the robustness of the proposed watermarking technology.

### 5 Experiments about Robustness

In the experiments, we select the gray Lena image with size 256×256 as host image and the encrypted face image as watermark. Using the above adaptive watermark embedding algorithm, we embed the encrypted watermark into the host image shown as Fig. 2. The PSNR of watermarked host image is 33.1358.



**Fig. 2.** Adaptive watermark embedding

To quantitatively evaluate the robustness, we processed the watermarked host image with additive noise, median filter and JPEG compression. Then the watermark information needs to be extracted and the *sim* value should be computed to detect whether the watermark exists. If the watermark exists, the extracted watermark needs to be decrypted. Results of the experiments are shown in Tab.1 and Fig.3.

**Table 1.** The results of experiments

Processing method	Processing intensity	PSNR	<i>sim</i>
Gaussian noise	Mean 0, Variance 0.0005	29.8779	0.8920
Salt-pepper noise	Mean 0, Variance 0.001	30.5550	0.9926
median filter	7×7	26.2942	0.5478
JPEG compression	Quality 70%	31.1711	0.7903



(a) Gaussian noise (b) Salt-pepper noise (c) Median filter (d) JPEG compression

**Fig. 3.** Decrypted result of the extracted watermark

Setting the threshold, we can detect the watermark’s existence under the condition of processing method and intensity mentioned in Tab.1. In Fig.3, the decrypted watermarks, except the one from the image processed with median filter, may be faintly distinguished. Therefore, the proposed watermarking algorithm can withstand the processing of adding noise, median filter and JPEG compression to a certain extent, demonstrating needful robustness.

## 6 Conclusion

The digital watermark satisfies some basic requirements for security, robustness, imperceptibility and authorization. In this paper, we adopt some technologies to satisfy these requirements:

(1) Based on the analysis of chaotic encryption methods, we research the two-dimensional Logistic map system and the chaotic conditions. After a key is selected, the chaotic map system is used to encrypt the meaningful gray image. This encryption method meets the requirement and has good secure performance.

(2) The proposed watermarking scheme that embeds the encrypted watermark into the DWT domain of host image is a digital watermarking technique in frequency domain. From the theory of the frequency domain watermark and the results of our experiment, this watermarking scheme demonstrates good robustness and can resist the attacks of the noise, filter and compression.

(3) Combined with HVS model, the embedding intensities for each sub-block are computed according to the characters of texture and energy of the host image. The watermark is self-adaptively embedded into the host image, which balances the requirements between robustness and imperceptibility.

(4) The correlation analysis can determine whether the appointed characteristic information exists in the pending image.

The grayscale face image and Lena image are used as examples in our experiments. In fact, other grayscale images or color images which intensity dealt as gray could be used and the same results can be gotten.

## References

- [1] Cox I J, Miller M L, Bloom J A. Digital watermarking. San Francisco: Morgan Kaufmann Publishers, 2002
- [2] Yang J, Lee M H, Chen X H et al. Mixing chaotic watermarks for embedding in wavelet transform domain. In: Proceedings of IEEE International Symposium on Circuits and Systems V2, Phoenix, USA, 2002:668-671
- [3] Yen J C. Watermarks embedded in the permuted image. In: Proceedings of IEEE International Symposium on Circuits and Systems V2, Sydney, Australia, 2001:53-56
- [4] Tefas A, Pitas I. Image authentication using chaotic mixing system. In: Proceedings of IEEE International Symposium on Circuits and Systems V1, Geneva, Switzerland, 2000:216-219
- [5] Ferretti A, Rahman N K. A study of coupled Logistic map and applications in chemical physics. *Chemical Physics*, 1988, 119:275-188
- [6] Nill N B. A visual model weighted cosine transform for image compression and quality assessment. *IEEE Transaction on Communication*, 1985, 33(6):551-557
- [7] Tan S H, Ngan K N. Classified perceptual coding with adaptive quantization. *IEEE Transaction on Circuits and Systems for Video Technology*, 1996, 37(6):375-383
- [8] NILL N B. A visual model weighted cosine transform for image compression and quality assessment. *IEEE Transaction on Communication*, 1985, 33(6):551-557
- [9] AT&T Laboratories Cambridge. The ORL Database of Faces. <http://www.uk.research.att.com/facedatabase.html>. 2004-11-12



# Identity-Based Universal Designated Verifier Signatures\*

Fanguo Zhang<sup>1</sup>, Willy Susilo<sup>2</sup>, Yi Mu<sup>2</sup>, and Xiaofeng Chen<sup>3</sup>

<sup>1</sup> Department of Electronics and Communication Engineering,  
Sun Yat-sen University, Guangzhou 510275, P.R. China  
isdzhfg@zsu.edu.cn

<sup>2</sup> School of Information Technology and Computer Science,  
University of Wollongong, Australia  
{wsusilo, ymu}@uow.edu.au

<sup>3</sup> Department of Computer Science,  
Sun Yat-sen University, Guangzhou 510275, P.R. China  
isschxf@zsu.edu.cn

**Abstract.** The notion of Universal Designated Verifier Signatures (UDVS) was introduced in the seminal paper of Steinfeld *et. al.* in [6]. In this paper, we firstly propose a model of identity-based (ID-based) UDVS schemes. We note that there are two methods to achieve an ID-based UDVS scheme. We provide two constructions of ID-based UDVS schemes based on bilinear pairings that use the two methods that we have identified. We provide our security proof based on the random oracle model.

## 1 Introduction

In a certificate-based public key system, before a user's public key is used, the participants must firstly verify the user's certificate. As a consequence, this system requires a large storage and computing time to store and verify each user's public key and the corresponding certificate. In 1984, Shamir [5] proposed ID-based cryptosystem to simplify key management procedures in certificate-based public key setting. Since then, many ID-based encryption and signature schemes have been proposed. The main idea of ID-based cryptosystems is that the identity information of each user serves as his/her public key.

In [6], Steinfeld *et. al.* proposed a special type of digital signature scheme called *Universal Designated Verifier Signatures* (UDVS), which directly addresses the user privacy issue in user certification systems. On one hand, UDVS scheme protects user's privacy, and on the other hand, it maintains a similar convenience of use for the user and for the certificate issuer CA as in certification systems using standard digital signatures. The scenario of UDVS schemes is as follows. A user Alice is issued a signed certificate by the CA. When Alice wishes to send her certificate to a verifier Bob, she uses Bob's public key to *transform* the CA's signature

---

\* This work is supported by the National Natural Science Foundation of China (No. 60403007) and ARC Discovery Grant DP0557493.

into a designated signature for Bob, using the UDVS scheme’s designation algorithm, and sends the transformed CA’s signature to Bob. Bob can use the CA’s public key to verify the designated signature on the certificate, but is unable to use this designated signature to convince any other third party that the certificate was indeed signed by the CA, even if Bob is willing to reveal his secret-key to the third party. This is achieved because Bob’s secret-key allows him to forge designated signatures by himself, so the third party is unable to tell who produced the signature (whereas Bob can, because he knows that he did not produce it). Therefore, through the use of a UDVS scheme, Alice’s privacy is preserved in the sense that Bob is unable to disseminate convincing statements about Alice (of course, nothing prevents Bob from revealing the certificate statements themselves to any third party, but the third party will be unable to tell whether these statements are authentic, i.e. whether they have been signed by the CA or not). A question that directly arises from this model is “how could one design an ID-based UDVS scheme that allows Alice to convince Bob, by only knowing Bob’s identity, such as email address, IP, etc.”?

### Our Contribution

In this paper, firstly we introduce the notion of ID-based UDVS schemes. We provide a model for such schemes together with its security requirements. We also propose two concrete constructions of ID-based UDVS schemes.

## 2 Preliminaries

### 2.1 Bilinear Pairings

Let  $\mathbb{G}_1$  be a cyclic additive group generated by  $P$ , whose order is a prime  $q$ , and  $\mathbb{G}_2$  be a cyclic multiplicative group with the same order  $q$ . Let  $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$  be a map with the following properties:

1. **Bilinearity:**  $e(aP, bQ) = e(P, Q)^{ab}$  for all  $P, Q \in \mathbb{G}_1, a, b \in \mathbb{Z}_q$
2. **Non-degeneracy:** There exists  $P, Q \in \mathbb{G}_1$  such that  $e(P, Q) \neq 1$ , in other words, the map does not send all pairs in  $\mathbb{G}_1 \times \mathbb{G}_1$  to the identity in  $\mathbb{G}_2$ ;
3. **Computability:** There is an efficient algorithm to compute  $e(P, Q)$  for all  $P, Q \in \mathbb{G}_1$ .

In our setting of prime order groups, the **Non-degeneracy** property is equivalent to  $e(P, Q) \neq 1$  for all  $P, Q \in \mathbb{G}_1$ . So, when  $P$  is a generator of  $\mathbb{G}_1$ ,  $e(P, P)$  is a generator of  $\mathbb{G}_2$ .

#### Definition 1. Bilinear Diffie-Hellman (BDH) Problem:

*Given a randomly chosen  $P \in \mathbb{G}_1$ , as well as  $aP, bP$  and  $cP$  (for unknown randomly chosen  $a, b, c \in \mathbb{Z}_q$ ), compute  $e(P, P)^{abc}$ .*

For the BDH problem to be hard,  $\mathbb{G}_1$  and  $\mathbb{G}_2$  must be chosen so that there is no known algorithm for efficiently solving the Diffie-Hellman problem in either  $\mathbb{G}_1$  or  $\mathbb{G}_2$ . We note that if the BDH problem is hard for a pairing  $e$ , then it follows that  $e$  is non-degenerate.

**Definition 2. Bilinear Diffie-Hellman Assumption:**

If  $\mathcal{IG}$  is a BDH parameter generator, the advantage  $\text{Adv}_{\mathcal{IG}}(\mathcal{A})$  that an algorithm  $\mathcal{A}$  has in solving the BDH problem is defined to be the probability that the algorithm  $\mathcal{A}$  outputs  $e(P, P)^{abc}$  on inputs  $\mathbb{G}_1, \mathbb{G}_2, e, P, aP, bP, cP$ , where  $(\mathbb{G}_1, \mathbb{G}_2, e)$  is the output of  $\mathcal{IG}$  for sufficiently large security parameter  $k$ ,  $P$  is a random generator of  $\mathbb{G}_1$  and  $a, b, c$  are random elements of  $\mathbb{Z}_q$ . The BDH assumption is that  $\text{Adv}_{\mathcal{IG}}(\mathcal{A})$  is negligible for all efficient algorithms  $\mathcal{A}$ .

Throughout this paper, we define the system parameters in all schemes as follows: Let  $P$  be a generator of  $\mathbb{G}_1$  with order  $q$ . The bilinear pairing is given by  $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ . Define two cryptographic hash functions  $H_1 : \{0, 1\}^* \rightarrow \{0, 1\}^\lambda$ , in general,  $|q| \geq \lambda \geq 160$ , and  $H_0 : \{0, 1\}^* \rightarrow \mathbb{G}_1^*$ . Denote  $\text{PARAMS} = \{\mathbb{G}_1, \mathbb{G}_2, e, q, \lambda, P, H_0, H_1\}$ , and let  $|q|$  denote size of  $q$  in bits.

**2.2 ID-Based Chameleon Hash Functions**

A chameleon hash function is associated with a pair of public and private keys and has the following properties [4]: (1) Anyone who knows the public key can compute the associated hash function. (2) For people who do not have the knowledge of the trapdoor (i.e. the secret key), the hash function is collision resistant: it is infeasible to find two inputs which are mapped to the same output. (3) The trapdoor information’s holder can easily find collisions for every given input.

The idea of chameleon hashing has been recently extended in [1] to construct an identity-based chameleon hash. An ID-based chameleon hash scheme is defined by a family of efficiently computable algorithms (Setup, Extract, Hash, Forge).

A number of ID-based Chameleon hash functions have been proposed, following the first paper proposed in [1]. In the setting of any ID-based system, there is a trusted party PKG, who only exists to initialize the system. In the following, we will review an ID-based Chameleon hash function from bilinear pairings in [8]. The four computable algorithms are defined as follows.

- Setup. PKG chooses a random number  $s \in \mathbb{Z}_q^*$  and sets  $P_{pub} = sP$ . PKG publishes  $\text{PARAMS} = \{\mathbb{G}_1, \mathbb{G}_2, e, q, P, P_{pub}, H_0, H_1\}$ , and keeps  $s$  as the MASTER KEY, which is known only by the PKG.
- Extract. A user submits his identity information ID to PKG. PKG computes the user’s public key as  $Q_{ID} = H_0(ID)$ , and returns  $S_{ID} = sQ_{ID}$  to the user as his private key.
- Hash. Given a message  $m$ , choose a random element  $R$  from  $\mathbb{G}_1$ . Define the hash as  $\text{Hash}(ID, m, R) = e(R, P)e(H_1(m)H_0(ID), P_{pub})$ .
- Forge.  $\text{Forge}(ID, S_{ID}, m, R, m') = R' = (H_1(m) - H_1(m'))S_{ID} + R$ . One can verify that  $\text{Hash}(ID, m, R) \stackrel{?}{=} \text{Hash}(ID, m', R')$  holds with equality.

**3 ID-Based Universal Designated Verifier Signature Schemes**

An ID-based Universal Designated Verifier Signature scheme ID-UDVS consists of six algorithms, namely (Setup, Extract, Sign, Public Verification, Designation, Designated Verification). There are four parties involved in the scheme:

- a PKG: is a trusted party who executes two operations: system setup (**Setup**) and user's private key generation (**Extract**).
- a *signer*  $S$ : who issued an ID based signature to be given to a *signature holder*.
- a *signature holder*  $SH$ : is a party who has a valid signature provided by a *signer*.
- a *designated verifier*  $DV$ : is any third party whose ID is published. *Anyone* who obtains a signature signed by the *signer* can always designate this signature to any third party, and this third party is referred as the designated verifier. In our scenario, any signature holder  $SH$  (who does not have any access to the signer's secret key) can designate the original signer's signature to a designated verifier  $DV$ , such that  $DV$  can be convinced with the authenticity of the signature, but he cannot convince any other third party about this fact, since he can always generate such a signature by himself which is indistinguishable from the original one.

The six algorithms defined in ID-UDVS are as follows.

1. **Setup** is a probabilistic polynomial algorithm, run by the PKG, that takes a security parameter  $k$  and returns PARAMS (system parameters) and MASTER-KEY.
2. **Extract** is a probabilistic polynomial algorithm, run by the PKG, that takes as input PARAMS, MASTER-KEY, and an arbitrary  $ID \in \{0, 1\}^*$ . It returns a private key  $\mathcal{S}_{ID}$ . Here ID is the signer's identity and will be used as the signer's public key.
3. **Sign** is a probabilistic polynomial algorithm that is executed by the signer  $S$ . It takes PARAMS, a private key  $\mathcal{S}_{ID}$ , an identity  $ID_S$  corresponding to the secret key  $\mathcal{S}_{ID}$ , and a message  $m$ . The algorithm outputs a signature  $\sigma(m)$  for  $m$ .
4. **Public Verification** is a deterministic polynomial algorithm that takes PARAMS, an identity of the signer  $ID_S$ , a message  $m$  and its signature  $\sigma(m)$ , and outputs either **accept** or **reject** as the verification decision.
5. **Designation** is a deterministic polynomial algorithm that takes as input PARAMS, a message  $m$ , a valid signature on  $m$ ,  $\sigma(m)$ , and an identity of the designated verifier  $ID_{DV}$ , and outputs a designated signature  $\sigma'(m)$  for  $m$ .
6. **Designated Verification** is a deterministic polynomial-time algorithm that takes a message  $m$ , a designated signature  $\sigma'(m)$ , the identity of the signer  $S$ ,  $ID_S$ , and the secret key of the designated verifier  $\mathcal{S}_{DV}$  and outputs either **accept** or **reject**.

There are essentially two ways to achieve an ID-UDVS scheme. We note that these methods *do not* imply a generic construction of an ID-UDVS scheme.

1. *By incorporating the identity or public key of the designated verifier to encrypt the signature.* Using this mechanism, a signature holder can *encrypt* a signature that he has with the designated verifier's ID (or public key), such that only the designated verifier can be convinced with the authenticity of the message. This way, only the designated verifier can verify the authenticity of the signature. We call this method as an ID-UDVS *scheme with PK encryption*.

2. *By incorporating a chameleon hash function.* Using this mechanism, a signature holder uses a published chameleon hash function that is owned by the designated verifier. The designated verifier can be convinced with the authenticity of the signature, but no any other third party can, since the designated verifier can always generate another valid message signature pair by himself. We call this method as an ID-UDVS scheme with a Chameleon Hash.

In section 4 and 5, we provide two schemes that use the above two mechanisms.

### 3.1 Security Requirements

*Security Against Existential Forgery on Adaptively Chosen Message and ID Attacks.* We say an ID – UDVS scheme, which consists of six algorithms (Setup, Extract, Sign, Public Verification, Designation, Designated Verification), is *secure against existential forgery on adaptively chosen message and ID attacks* if no polynomial time algorithm  $\mathcal{A}$  has a non-negligible advantage against a challenger  $\mathcal{C}$  in the following game.

1.  $\mathcal{C}$  runs Setup of the scheme. The resulting PARAMS is given to  $\mathcal{A}$ . MASTER KEY is kept secret from  $\mathcal{A}$ .
2.  $\mathcal{A}$  issues the following queries as he wants.
  - (a) Extract query: Given an identity ID,  $\mathcal{C}$  returns the private key  $\mathcal{S}_{ID}$  corresponding to ID which is obtained by executing Extract.
  - (b) Sign query: Given an identity ID and a message  $m$ ,  $\mathcal{C}$  returns a signature  $\sigma(m)$  which is obtained by running Sign.
3.  $\mathcal{A}$  outputs  $(ID_S, ID_{DV}, m, \sigma'(m))$  where  $ID_S$  is the identity of a signer,  $ID_{DV}$  is the identity of a designated verifier,  $ID_S$  and  $ID_{DV}$  have never been queried to the Extract query and  $(ID_S, m)$  has never been queried before to the Sign query.  $\mathcal{A}$  wins the game if  $\sigma'(m)$  is a valid designated signature on  $m$ . That is,  $\text{DesignatedVerification}(m, \sigma'(m), ID_S, \mathcal{S}_{DV}) \stackrel{?}{=} \text{accept}$  holds with equality.

We define  $\mathcal{A}$ 's guessing advantage  $\text{Adv}_{ID-UDVS}(\mathcal{A}) = |\text{Pr}[\beta' = \beta] - \frac{1}{2}|$ .

## 4 An ID-UDVS Scheme with a PK Encryption from Bilinear Pairings

In this section, we provide our first construction of an ID-based UDVS (ID-UDVS) scheme based on bilinear pairings. Our ID-UDVS scheme functions as a standard Cha-Cheon signature [3] scheme when no designation is performed. Hence, it is compatible with the key generation, signing and verifying algorithms of the Cha-Cheon signature scheme [3]. The scheme is as follows.

1. Setup: PKG chooses a random number  $s \in Z_q^*$  and sets  $P_{pub} = sP$ . PKG publishes system parameters  $\text{PARAMS} = \{\mathbb{G}_1, \mathbb{G}_2, e, q, P, P_{pub}, H_0, H_1\}$ , and keeps  $s$  as the MASTER KEY, which is known only by itself.

2. **Extract:** A user submits his/her identity information  $ID$  to PKG. After a valid identification, PKG computes the user's public key as  $Q_{ID} = H_0(ID)$ , and returns  $S_{ID} = sQ_{ID}$  to the user as his/her private key.
3. **Sign.** Given a secret key  $S_{ID}$ , and a message  $m$ , perform the following.
  - Compute  $U = rQ_{ID}$ , where  $r \in_R \mathbb{Z}_q^*$ ,  $h = H_1(U||m)$
  - Compute  $V = (r + h)S_{ID}$ .
  - Output the signature on  $m$  as  $(U, V)$ .
4. **Public Verification.** Given  $ID$ , a message  $m$ , and a signature  $(U, V)$ , verify if

$$e(V, P) \stackrel{?}{=} e(U + H_1(U||m)Q_{ID}, P_{pub})$$

holds with equality. If so, then output **accept**. Otherwise, output **reject**.

5. **Designation.** Given the signer's public key  $ID$ , a verifier's public key  $ID_{DV}$  and a message-signature pair  $(m, U, V)$ , compute  $\sigma' = e(V, Q_{ID_{DV}})$ , where  $Q_{ID_{DV}} = H_0(ID_{DV})$ . The designated verifier signature is  $(U, \sigma')$ .
6. **Designated Verification.** Given the signer's public key  $ID$ , a verifier's secret key  $S_{ID_{DV}}$  and a message/designated signature pair  $(m, U, \sigma')$ , **accept** if and only if

$$e(U + H_1(U||m)Q_{ID}, S_{ID_{DV}}) \stackrel{?}{=} \sigma'$$

holds with equality. Otherwise, output **reject**.

### 4.1 Security Analysis

#### *Correctness and Consistency.*

The correctness and consistency of the scheme is justified as follows.

$$\begin{aligned} e(V, P) &= e((r + h)S_{ID}, P) = e((r + h)sQ_{ID}, P) \\ &= e((r + h)Q_{ID}, P_{pub}) = e(rQ_{ID}, P_{pub})e(hQ_{ID}, P_{pub}) \\ &= e(U, P_{pub})e(H_1(U||m)Q_{ID}, P_{pub}) = e(U + H_1(U||m)Q_{ID}, P_{pub}) \end{aligned}$$

$$\begin{aligned} e(U + H_1(U||m)Q_{ID}, S_{ID_{DV}}) &= e(rQ_{ID} + H_1(U||m)Q_{ID}, sQ_{ID_{DV}}) \\ &= e((r + h)sQ_{ID}, Q_{ID_{DV}}) = e((r + h)S_{ID}, Q_{ID_{DV}}) = e(V, Q_{ID_{DV}}) = \sigma' \end{aligned}$$

**Theorem 1.** *If a valid universal designated signature can be generated without the knowledge of a valid signature or a secret key of the signer, then the BDH problem may be solved in a polynomial time.*

*Proof.* Let us recall the BDH problem as follows. Given a randomly chosen  $P \in \mathbb{G}_1$ , as well as  $aP, bP$  and  $cP$  (for unknown randomly chosen  $a, b, c \in \mathbb{Z}_q$ ), compute  $e(P, P)^{abc}$ . To show the proof, we assume there is a polynomial algorithm  $\mathcal{A}$  that can generate a valid universal designated signature  $\sigma'$  for a message  $m$ , without the knowledge of a signature  $\sigma$  generated by the signer, and without the signer's secret key. The algorithm  $\mathcal{A}$  accepts an  $ID$  of the signer,

$ID_A$ , an ID of the designated verifier,  $ID_C$ , and a message  $m$ , and it outputs a valid universal designated signature  $(U, \sigma')$ , where

$$Pr[\text{DesignatedVerification}(m, (U, \sigma'), ID_C) = \text{accept}] = 1.$$

We will show how to use this algorithm to solve the BDH problem.

In our setting, we know the public information  $P_{pub}, ID_A$  (the ID of the signer) and  $ID_C$  (the ID of the designated verifier). From this public information, we can obtain  $Q_{ID_A} = H_0(ID_A)$  and  $Q_{ID_C} = H_0(ID_C)$ . Since  $P$  is a generator in  $\mathbb{G}_1$ , then we can rewrite these three parameters as

$$Q_{ID_A} = aP \quad P_{pub} = bP \quad Q_{ID_C} = cP$$

We note that  $S_{ID_A} = bQ_{ID_A} = abP$  and  $S_{ID_C} = bQ_{ID_C} = bcP$ . Now, we construct an algorithm  $\hat{A}$  to solve the BDH problem as follows. Algorithm  $\hat{A}$  will control  $\mathcal{A}$  and replaces  $\mathcal{A}$ 's interaction with the signer by simulation. Firstly,  $\hat{A}$  generates a list of ID of its choice, together with a random  $s_i$  associated with it. The size of this set is  $2^\ell$ , where  $\ell$  is the security parameter. The idea of the game is illustrated as follows. The purpose of  $\hat{A}$  is to inject the information above  $(aP, bP, cP)$  during the simulation. Without losing generality, we only show the interaction where  $\mathcal{A}$  interacts with  $\hat{A}$  for the information that  $\hat{A}$  wants. There is a probability that  $\hat{A}$  will fail, i.e. when  $\mathcal{A}$  queries the secret key for either  $ID_A$  or  $ID_C$  that will match with the published  $P_{pub}$ . Since  $\hat{A}$  does not have this information, then  $\hat{A}$  will halt the game. The probability of this failure to happen is  $\leq \frac{1}{2^\ell}$ .  $\mathcal{A}$  will be run twice with a different random query set, but from the same list of ID's generated at the first place. The attack is successful, when  $\mathcal{A}$  outputs two signatures for the given parameters (forking lemma). More concretely, the algorithm is described as follows. Firstly,  $\hat{A}$  selects two random numbers  $a', a'' \in \mathbb{Z}_q$ , where  $a' - a'' = 1 \pmod{q}$ . Then,  $\hat{A}$  will control  $\mathcal{A}$  as follows.

*First Round*

*H<sub>1</sub> - Hash Query.* When  $\mathcal{A}$  requests the value of  $H_1(U_1||m)$ , for the targeted parameters,  $\hat{A}$  responds with  $a'Q_{ID_A}$ . Otherwise, responds with the list that he has generated.

*Random Generation Query.* When  $\mathcal{A}$  requests  $\hat{A}$  to generate a random number  $r \in \mathbb{Z}_q$  and returns  $U$ , if the targeted parameters are used, then  $\hat{A}$  responds by  $r \in \mathbb{Z}_q$ , keeps this  $r$  in his separate list and returns  $rP$ .

*Output.* Eventually, the output of the first round is  $(U, \sigma'_1)$  where  $\sigma'_1 = e(U + a'Q_{ID_A}, S_{ID_C})$ .

*Second Round*

*H<sub>1</sub> - Hash Query.* When  $\mathcal{A}$  requests the value of  $H_1(U_1||m)$ , for the targeted parameters,  $\hat{A}$  responds with  $a''Q_{ID_A}$ . Otherwise, responds with the list that he has generated.

*Random Generation Query.* When  $\mathcal{A}$  requests  $\hat{A}$  to generate a random number  $r \in \mathbb{Z}_q$  and returns  $U$ , if the targeted parameters are used, then  $\hat{A}$  responds returning  $rP$ , where  $r$  is the number that he kept from the first round.

*Output.* Eventually, the output of the first round is  $(U, \sigma'_2)$  where  $\sigma'_2 = e(U + a''Q_{ID_A}, S_{ID_C})$ .

Obtaining  $(U, \sigma'_1)$  and  $(U, \sigma'_2)$ ,  $\hat{A}$  can solve the BDH problem by first computing  $d = \frac{\sigma'_1}{\sigma'_2}$  and output  $d$  as the solution of BDH problem.

The correctness of this algorithm is justified as follows.

$$\begin{aligned} d &= \frac{\sigma'_1}{\sigma'_2} = e(U + a'Q_{ID_A}, S_{ID_C}) / e(U + a''Q_{ID_A}, S_{ID_C}) = e((a' - a'')Q_{ID_A}, S_{ID_C}) \\ &= e(Q_{ID_A}, S_{ID_C}) = e(aP, bcP) = e(P, P)^{abc} \end{aligned}$$

This contradicts with the BDH assumption, and hence, we complete the proof. As mentioned earlier, the probability that the simulation will fail is  $\leq \frac{1}{2^\ell}$ , where  $\ell$  is the security parameter. ■

**Theorem 2.** *Having received a UDVS signature  $(U, \sigma')$ , the designated verifier DV cannot convince any other third party about the authenticity of the designated signature.*

*Proof.* The designated verifier DV cannot convince anyone else about the authenticity of  $(U, \sigma')$  because he can always generate this signature by himself after observing  $U$ . More precisely, he can always generate  $\hat{U} = rQ_{ID_A}$ , for a random  $r \in \mathbb{Z}_q$ , and compute  $\hat{\sigma}' = e(\hat{U} + H_1(\hat{U}||m')Q_{ID}, S_{ID_{DV}})$ , for a random  $m' \in \mathbb{Z}_q$ , where  $m' \neq m$ , which is indistinguishable from the original signature. We note that the new pair  $(\hat{U}, \hat{\sigma}')$  will pass the Designated Verification algorithm. ■

## 5 An ID-UDVS Scheme with a Chameleon Hash from Bilinear Pairings

In this section, we present our second ID-based UDVS scheme. In contrast to our first scheme, our second scheme makes use of bilinear pairings together with an ID-based chameleon hash function. The scheme is as follows.

- **Setup:** PKG selects a random number  $s \in \mathbb{Z}_q^*$  and sets  $P_{pub} = sP$ . Define another cryptographic hash function:  $H_1 : \{0, 1\}^* \rightarrow \mathbb{Z}_q^*$  and ID-Based Chameleon Hash: Hash. The center publishes system parameters  $PARAMS = \{\mathbb{G}_1, \mathbb{G}_2, e, q, P, P_{pub}, H_0, H_1\}$  and the ID-based Chameleon Hash Hash.
- **Extract:** A user submits his/her identity information ID to PKG. PKG computes the user's public key as  $Q_{ID} = H_0(ID)$ , and returns  $S_{ID} = sQ_{ID}$  to the user as his/her private key.
- **Sign.** Given a secret key  $S_{ID}$ , and a message  $m \in \mathbb{Z}_q$ , compute  $r = e(P, P)^k$ , where  $k \in_R \mathbb{Z}_q^*$ ,  $c = H_1(m||r)$  and  $U = kP - cS_{ID}$ . The signature on a message  $m$  is  $\sigma = (c, U)$ .
- **Public Verification.** Given ID, a message  $m$ , and a signature  $(c, U)$ , verify if

$$c \stackrel{?}{=} H_1(m||e(U, P)e(Q_{ID}, P_{pub})^c)$$

holds with equality.



- **Designation.** Given the signer’s public key  $ID$ , a verifier’s public key  $ID_{DV}$  and a message/signature pair  $(m, c, U)$ , create a UDVS signature as follows.
  - Compute  $r = e(U, P)e(Q_{ID}, P_{pub})^c$ .
  - Compute  $r' = H_1(e(P, P)^{k'})$  for a random  $k' \in \mathbb{Z}_q^*$ .
  - Compute  $h = \text{Hash}(ID_{DV}, r', R)$  for a random  $R \in \mathbb{G}_1$ .
  - Compute  $c' = H_1(m, c, r, h)$ .
  - Compute  $S' = k'P - c'U$ .
  - Output the designated signature as  $\sigma' = (r, R, c', S')$ .
- **Designated Verification.** Given the signer’s public key  $ID$ , a verifier’s secret key  $S_{ID_{DV}}$  and a message/UDVS signature pair  $m, (r, R, c', S')$ , accept if and only if

$$c' \stackrel{?}{=} H_1(m, c, r, h)$$

holds with equality. Here,  $c = H_1(m||r)$ ,  $h = \text{Hash}(ID_{DV}, R, r')$ , and  $r' = H_1(e(S', P)(r \cdot e(Q_{ID}, P_{pub})^{-c})^{c'})$ .

### 5.1 Security Analysis

#### *Correctness and Consistency*

The correctness and consistency of our second scheme are justified as follows.

$$\begin{aligned} c &= H_1(m||e(U, P)e(Q_{ID}, P_{pub})^c) = H_1(m||e(kP - cS_{ID}, P)e(Q_{ID}, P_{pub})^c) \\ &= H_1(m||e(P, P)^k e(cS_{ID}, P)e(cS_{ID}, P)^{-1}) = H_1(m||e(P, P)^k) = H_1(m||r) \end{aligned}$$

It is easy to see that the following equation holds with equality when it is generated correctly:  $c' \stackrel{?}{=} H_1(m, c, r, h)$  for  $c = H_1(m||r)$ ,  $h = \text{Hash}(ID_{DV}, R, r')$ , and  $r' = H_1(e(S', P)(r \cdot e(Q_{ID}, P_{pub})^{-c})^{c'})$ . ■

**Theorem 3.** *The designated signature  $(r, R, c', S')$  on a message  $m$  cannot be used by the designated verifier DV to convince any other third party.*

*Proof.* DV can always generate the designated verifier  $(r, R, c', S')$  on a message  $m' \in \mathbb{Z}_q$ , where  $m' \neq m$ , by himself, which is indistinguishable from the original signature. The way to do this is as follows.

- Select a random message  $m' \in \mathbb{Z}_q$ , and a random number  $r \in \mathbb{Z}_q$ .
- Compute  $c = H_1(m' || r)$ .
- Compute  $r' = e(P, P)^{k'}$ , for a random  $k' \in \mathbb{Z}_q$ .
- Compute  $h = \text{Hash}(ID_{DV}, R, r')$ , for a random  $R \in \mathbb{G}_1$ .
- Compute  $S' = k'P - c'U$ .
- Output  $(r, R, c', S')$ .

Moreover, after receiving a valid designated signature  $(r, R, c', S')$ , the designated signature still can modify this signature by executing the `Forge` algorithm. Due to the construction of the ID-based Chameleon Hash function used, he can always find a different  $R' \neq R$  that will satisfy the `DesignatedVerification` algorithm. ■

The formal security proof is omitted due to page limitation.

## 6 Conclusion

In this paper, we propose a formal definition for identity-based Universal Designated Verifier Signatures (ID-UDVS). We provide two secure ID-UDVS schemes based on bilinear pairings. Our first scheme uses the Cha-Cheon ID-based signature scheme, while our second scheme uses an ID-based Chameleon Hash function.

## References

1. G. Ateniese and B. de Medeiros. Identity-based Chameleon Hash and Applications. *Financial Cryptography 2004, LNCS 3110*, pages 164 - 180, 2004.
2. D. Boneh, B. Lynn, and H. Shacham. Short signatures from the weil pairing. *Advanced in Cryptology - Asiacrypt 2001, LNCS 2248*, pages 514–532, 2001.
3. J. C. Cha and J. H. Cheon. An Identity-Based Signature from Gap Diffie-Hellman Groups. *6th International Workshop on Theory and Practice in PKC (PKC 2003), LNCS 2567*, pages 18–30, 2003.
4. H. Krawczyk and T. Rabin. Chameleon hashing and signatures. *Network and Distr System Security Symp, The Internet Society*, pages 143 – 154, 2000.
5. A. Shamir. Identity-based cryptosystems and signature schemes. *Advances in Cryptology - Crypto '84, LNCS 196*, pages 47–53, 1985.
6. R. Steinfeld, L. Bull, H. Wang, and J. Pieprzyk. Universal designated-verifier signatures. *Advances in Cryptology - Asiacrypt 2003, LNCS 2894*, pages 523 – 543, 2003.
7. R. Steinfeld, H. Wang, and J. Pieprzyk. Efficient Extension of Standard Schnorr/RSA signatures into Universal Designated-Verifier Signatures. *7th International Workshop on Theory and Practice in PKC (PKC 2004), LNCS 2947*, pages 86–100, 2004.
8. F. Zhang, R. Safavi-Naini, and W. Susilo. ID-Based Chameleon Hashes from Bilinear Pairings. *Cryptology ePrint Archive, Report 2003/208*, 2003.

# Short Designated Verifier Proxy Signature from Pairings<sup>\*</sup>

Xinyi Huang<sup>1</sup>, Yi Mu<sup>2</sup>, Willy Susilo<sup>2</sup>, and Futai Zhang<sup>1,\*\*</sup>

<sup>1</sup> College of Mathematics and Computer Science,  
Nanjing Normal University, P.R. China  
xinyinjnu@126.com, zhangfutai@njnu.edu.cn

<sup>2</sup> Centre for Information Security Research,  
School of Information Technology and Computer Science,  
University of Wollongong, Australia  
{wsusilo, ymu}@uow.edu.au

**Abstract.** In a designated verifier proxy signature scheme, the original signer delegates her/his signing capability to the proxy signer in such a way that the latter can sign messages on behalf of the former, but only the designated verifier can believe the validity of these signatures. In this paper, we firstly describe the notion of short designated verifier proxy signature, which we call SDVPS. Then a concrete scheme is presented. We prove that the proposed scheme is unforgeable even to the original signer under the Gap Bilinear Diffie-Hellman assumption and Random Oracle Model.

**Keywords:** Proxy Signature, Short Signature, Pairings, Authentication.

## 1 Introduction

In a proxy signature scheme, the original signer (say, Alice) can delegate her signing right to another user (say, Bob) who is called proxy signer. Bob can sign messages on behalf of Alice. Upon receiving a proxy signature on some message, the verifier can validate its correctness by a given verification procedure and can be convinced of the original signer's agreement on the proxy signing. The notion of proxy signature was introduced in [7]. Proxy signature schemes have been suggested for use in a number of applications, including electronic commerce and distributed shared object systems. Based on the application, they can be classified as *full delegation*, *partial delegation*, and *delegation by warrant* schemes. Based on the knowledge of the proxy private key, proxy signatures can be classified into *proxy-unprotected* and *proxy-protected*. In a proxy-protected scheme only the proxy signer can generate proxy signatures, while in a proxy-unprotected scheme either the proxy signer or the original signer can generate

---

<sup>\*</sup> This work is supported by ARC Discovery Grant DP0557493.

<sup>\*\*</sup> Partially supported by Ministry of Education of Jiangsu Province Project 03KJA520066 and Open Project of Key Laboratory on Computer Network and Information Security of Ministry of Education of China.

proxy signatures since both of them have a knowledge on the proxy private key. In many applications, proxy-protected schemes are required to avoid the potential disputes between the original signer and the proxy signer.

There have been several interesting works that provide different features to proxy signature, for example, threshold proxy signature [15], one-time proxy signature [13], ID-based proxy signature [14], etc. Let's consider a scenario where the proxy signer wishes to protect his signing privilege from knowing by other parties. That is, Bob only wants to convince the designated receiver that he has signed the specific message. This scenario is related to the *designated verifier* signatures proposed by Jakobsson, Sako and Impagliazzo in [4]. This signature scheme can be considered as the first non-interactive undeniable signature scheme that transforms Chaum's scheme [1] into non-interactive verification using a designated verifier proof. In a designated verifier scheme, the signature provides authentication of a message without providing a non-repudiation property of traditional signatures. A designated verifier scheme can be used to convince a single third party, i.e., the designated verifier, and only the designated verifier can be convinced about its validity or invalidity. This is due to the fact that the designated verifier can always create a signature intended for himself that is indistinguishable from an original signature. This scheme does not require any interaction with the presumed signer to verify the authenticity of the message. There are a number of other works on designated verifier signatures, for example [5, 4, 9, 10, 8, 11].

Constructing an ordinary designated verifier proxy signature scheme is trivial (e.g., [2],[12]). The motivation of this paper is to find a scheme of designated verifier proxy signature which is *very short*. We call it Short Designated Verifier Proxy Signature (SDVPS). Compared with other schemes, our proxy key generation is *noninteractive* and the signature length is *shortest*. We prove that our scheme is proxy-protected that is even the original signer cannot forge a valid signature. The proof is based on the Gap Bilinear Diffie-Hellman problem in random oracle.

The rest of this paper is organized as follows. In the next section, we will provide some preliminaries and background required throughout the paper. In Section 3, we introduce the notion of the SDVPS scheme. In Section 4, we provide our concrete SDVPS scheme, and its security proof is given in Section 5. In Section 6, we compare the performance of our scheme with the existing scheme. Section 7 concludes this paper.

## 2 Preliminaries

In this section, we will review some fundamental backgrounds required in this paper, namely bilinear pairing and the definition of the designated verifier signature.

### 2.1 Basic Concepts on Bilinear Pairings

Let  $\mathbb{G}_1, \mathbb{G}_2$  be cyclic additive groups generated by  $P_1, P_2$ , respectively, whose orders are a prime  $q$ . Let  $\mathbb{G}_M$  be a cyclic multiplicative group with the same

order  $q$ . We assume there is an isomorphism  $\psi : \mathbb{G}_2 \rightarrow \mathbb{G}_1$  such that  $\psi(P_2) = P_1$ . Let  $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_M$  be a bilinear mapping with the following properties:

1. *Bilinearity*:  $e(aP, bQ) = e(P, Q)^{ab}$  for all  $P \in \mathbb{G}_1, Q \in \mathbb{G}_2, a, b, \in \mathbb{Z}_q$ .
2. *Non-degeneracy*: There exists  $P \in \mathbb{G}_1, Q \in \mathbb{G}_2$  such that  $e(P, Q) \neq 1_{\mathbb{G}_M}$ .
3. *Computability*: There exists an efficient algorithm to compute  $e(P, Q)$  for all  $P \in \mathbb{G}_1, Q \in \mathbb{G}_2$ .

For simplicity, hereafter, we set  $\mathbb{G}_1 = \mathbb{G}_2$  and  $P_1 = P_2$ . We note that our scheme can be easily modified for a general case, when  $\mathbb{G}_1 \neq \mathbb{G}_2$ .

## 2.2 Complexity Assumptions

We assume that the Bilinear Diffie-Hellman problem is intractable in polynomial time. Formally, we define it as follows.

### Definition 1. Bilinear Diffie-Hellman (BDH) Problem

Given a randomly chosen  $P \in \mathbb{G}_1$ , as well as  $aP, bP$  and  $cP$  (for unknown randomly chosen  $a, b, c \in \mathbb{Z}_q^*$ ), compute  $e(P, P)^{abc}$ .

### Definition 2. Decisional Bilinear Diffie-Hellman (DBDH) Problem

Given a randomly chosen  $P \in \mathbb{G}_1$ , as well as  $aP, bP, cP$  (for unknown randomly chosen  $a, b, c \in \mathbb{Z}_q^*$ ) and  $h \in \mathbb{G}_M$ , decide whether  $h = e(P, P)^{abc}$ .

### Definition 3. Gap Bilinear Diffie-Hellman (GBDH) Problem

Given a randomly chosen  $P \in \mathbb{G}_1$ , as well as  $aP, bP$  and  $cP$  (for unknown randomly chosen  $a, b, c \in \mathbb{Z}_q^*$ ), compute  $e(P, P)^{abc}$  with the help of the DBDH oracle.

## 2.3 Designated Verifier Signature

The goal of designated verifier proofs is to allow an entity, Alice, to prove the validity of a statement  $\Theta$  to a specific entity, Bob, in such a way that Bob is convinced about this fact but he cannot transfer this conviction to other third party. In [4], it is suggested that Alice should prove the statement “ $\Theta$  is correct or I know Bob’s secret key”. Bob, who is aware that he has not generated the proof himself and also sure that Alice does not know his secret key will be convinced by this proof (i.e. the first part of the proof, namely  $\Theta$  is correct), while no other verifier can decide which part of the disjunction is correct.

The notion of designated verifier proofs are given in [4], and they are formalized in [8] as follows.

### Definition 1. Designated Verifier Signature [8]

Let  $P(A, B)$  be a protocol between Alice and Bob so that Alice can prove the correctness of statement  $\Theta$ . Bob is said to be a designated verifier if he can produce identically distributed transcripts that are indistinguishable from those of  $P(A, B)$ .

### 3 Short Designated Verifier Proxy Signature(SDVPS)

#### 3.1 Outline of the SDVPS

There exist three participants in the system, namely Alice, Bob and Cindy, who act as the original signer, the proxy signer and the receiver (or the designated verifier), respectively. We denote  $(x_i, P_i)$  as a pair of private key and public key for user  $i$ , where  $i \in \{A, B, C\}$  indicating Alice, Bob, and Cindy, respectively. A short designated verifier proxy signature scheme (SDVPS) consists of following six essential algorithms:

- **ParamGen**: It takes as input the system security parameter  $\ell$  and outputs the system parameters.
- **KeyGen**: It takes as input the security parameter  $\ell$  and outputs the key set:  $(x_i, P_i)$  for  $i = A, B, C$ .
- **ProxyKeyGen**: A deterministic algorithm that takes as input the original signer's secret key, the proxy signer's secret key, the identity of the proxy signer and the warrant  $m_w$  to generate the proxykey. That is  $\text{proxykey} \leftarrow \text{ProxyKeyGen}(x_A, x_B, ID_B, m_w)$ . where  $x_A, x_B$  is the secret key of the original signer and the proxy signer,  $ID_B$  is the identity of the proxy signer.
- **Sign**: A deterministic algorithm that takes as input the proxykey, the designated verifier's public key and a message  $m$  to generate a signature  $\sigma$ . That is  $\sigma \leftarrow \text{Sign}(\text{proxykey}, ID_B, P_C, m)$ , where  $\text{proxykey}$  is generated by the above **ProxyKeyGen** algorithm,  $ID_B$  is the identity of the proxy signer and  $P_C$  is the public key of the receiver(the designated verifier).
- **Verify**: A deterministic algorithm that accepts a message  $m$ , a signature  $\sigma$ , the original signer's public key  $P_A$ , the proxy signer's public key  $P_B$ , the proxy signer's identity and the receiver's secret key  $x_C$  and returns **True** if the signature is correct, or  $\perp$  otherwise. That is,  $\{\text{True}, \perp\} \leftarrow \text{Verify}(P_A, P_B, ID_B, x_C, m, \sigma)$ .
- **Transcript Simulation**: An algorithm that is run by the verifier to produce identically distributed transcripts that are *indistinguishable* from the original protocol.

In addition to the above main algorithms, we also require the following.

- **Correctness**. All signatures generated correctly by **Sign** algorithm must always pass the verification algorithm. That is,

$$\Pr(\text{True} \leftarrow \text{Verify}(P_A, P_B, ID_B, x_C, m, \text{Sign}(\text{proxykey}, ID_B, P_C, m), m_w)) = 1.$$

- **Transcript Simulation Generation**. We require that the verifier, who holds the secret key  $x_C$  can always produce identically distributed transcripts that are indistinguishable from the original protocol via the **Transcript Simulation** algorithm.

### 3.2 Security Model

There are three types adversaries in the system:

1. **Type I:** This type adversary only has the public keys of Alice and Bob.
2. **Type II:** This type of adversary has the public keys of Alice and Bob, her/he also has the secret key of Bob (the proxy signer).
3. **Type III:** This type of adversary has the public keys of Alice and Bob, her/he also has the secret key of Alice (the original signer).

We can find that if our short proxy signature scheme is unforgeable against Type II (or Type III) adversary, our scheme is also unforgeable against Type I adversary.

#### Formal Security Notion: Unforgeability of the SDVPS

We provide a formal definition of existential unforgeability of a short designated verifier proxy signature scheme (SDVPS) under a chosen message attack (EF-CMA-adversary). It is defined using the following game between an adversary  $\mathcal{A}$  and a challenger  $\mathcal{C}$ .

- **Setup:**  $\mathcal{C}$  runs the algorithm to generate the public keys ( $P_A, P_B$  and  $P_C$ ) of the original signer  $A$ , proxy signer  $B$  and the designated verifier  $C$ .  $\mathcal{C}$  also generates the identity  $ID_B$  of the proxy signer.
- **Sign Queries:**  $\mathcal{A}$  can request a proxy signature on a message  $m$  with the original signer  $A$ , the proxy signer  $B$  and the designated verifier  $C$ . In response,  $\mathcal{C}$  outputs a signature  $\sigma$  for a message  $m$ .
- **Verify Queries:**  $\mathcal{A}$  can request a signature verification on a pair  $(m, \sigma)$  with the original signer  $A$ , the proxy signer  $B$  and the designated verifier  $C$ . In response,  $\mathcal{C}$  outputs **True** if it is correct, or  $\perp$  otherwise.
- **Output:** Finally,  $\mathcal{A}$  outputs a new pair  $(m^*, \sigma^*)$ , where  $m^*$  has never been queried during the **Sign Queries** and  $\sigma^*$  is a valid signature for the original signer  $A$ , the proxy signer  $B$  and the designated verifier  $C$ .

The success probability of an adversary to win the game is defined by

$$Succ_{SDVPS, \mathcal{A}}^{EF-CMA}(\ell).$$

**Definition 4.** We say that a short designated verifier proxy signature scheme is existentially unforgeable under a chosen message attack if the probability of success of any polynomially bounded adversary in the above game is negligible for all the three types of adversaries. In other words,  $Succ_{SDVPS, \mathcal{A}}^{EF-CMA}(\ell) \leq \epsilon$  where  $\mathcal{A} \in \{\mathcal{A}_I, \mathcal{A}_{II}, \mathcal{A}_{III}\}$  and  $\epsilon$  is negligible.

## 4 Our SDVPS Scheme

As assumed earlier, there are three participants in the system, namely Alice, Bob and Cindy, who act as the original signer, the proxy signer and the receiver (or the designated verifier), respectively. Our SDVPS consists of the following algorithms.

1. **ParamGen**: Taking as input the system security parameter  $\ell$ , the algorithm outputs  $\{\mathbb{G}_1, \mathbb{G}_M, q, e, P\}$ , including a cyclic additive group  $\mathbb{G}_1$  of order  $q (q \geq 2^\ell)$ , a multiplicative group  $\mathbb{G}_M$  of order  $q$ , a bilinear map  $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_M$  and a generator  $P$  of  $\mathbb{G}_1$ . This algorithm also outputs two cryptographic hash functions  $H_0$  and  $H_1$  where  $H_0 : \{0, 1\}^* \rightarrow \mathbb{G}_1$  and  $H_1 : \{0, 1\}^* \rightarrow \mathbb{Z}_q^*$ .
2. **KeyGen**: Taking as input the system security parameter  $k$ , the algorithm outputs three pairs of secret/public keys  $(x_i, P_i = x_i P)$ , for  $i = A, B, C$ , which denote Alice, Bob, and Cindy, respectively.
3. **ProxyKeyGen**:
  - (a) Alice computes  $D_{AB} = x_A Q_B$ , where  $Q_B = H_0(ID_B, P_B, m_w)$ ,  $ID_B$  is the identity of Bob,  $P_B$  is the public key of Bob, and  $m_w$  is the warrant. Alice then sends  $(D_{AB}, m_w)$  to Bob.
  - (b) Bob verifies whether  $e(D_{AB}, P) = e(Q_B, P_A)$  holds.
  - (c) Bob obtains the proxykey  $(x_B, D_{AB})$ .
4. **Sign**: For a message  $m$ , Bob computes  $\sigma = H_1(m, e(D_{AB} + x_B Q_B, P_C))$  and the designated verifier proxy signature on the message  $m$  is  $\sigma$ .
5. **Verify**: To check whether  $\sigma$  is a valid signature of the message  $m$  and the warrant  $m_w$ , Cindy uses her secret key  $x_C$  to check:  $\sigma \stackrel{?}{=} H_1(m, e(x_C Q_B, P_A + P_B))$  where  $Q_B = H_0(ID_B, P_B, m_w)$ . If the above equation holds, Cindy accepts the signature  $\sigma$ , otherwise rejects it.

Correctness:

$$\begin{aligned} H_1(m, e(x_C Q_B, P_A + P_B)) &= H_1(m, e(x_C Q_B, x_A P + x_B P)) \\ &= H_1(m, e((x_A + x_B) Q_B, x_C P)) = H_1(m, e(D_{AB} + x_B Q_B, P_C)) \end{aligned}$$

Transcript Simulation:

Cindy can use her secret key to compute an arbitrary signature on a message  $m^*$  as  $\sigma^* = H_1(m^*, e(x_C Q_B, P_A + P_B))$ .

## 5 Security Analysis

In this section, we will firstly prove that the proposed scheme is a designated verifier signature scheme. Then we prove that our SDVPS is secure against all types of adversaries.

**Theorem 1.** *The proposed scheme is a designated verifier signature scheme.*

*Proof:* For any message  $m$ , Cindy can compute a valid signature by computing  $\sigma = H_1(m, e(x_C Q_B, P_A + P_B))$ . One can find that signature generated like this is the same as the original one generated by the proxy signer Bob. Therefore, even given Cindy's secret key  $x_C$ , no one can believe the signature is sent by Bob.

**Theorem 2.** *If the Type II Adversary  $\mathcal{A}_{II}$  (the proxy signer Bob) can forge a valid signature of the proposed scheme with success probability  $\text{Succ}_{SDVPS, \mathcal{A}_{II}}^{EF-CMA}$  after making  $q_H$  queries to the  $H_1 : \{0, 1\}^* \rightarrow \mathbb{Z}_q^*$  ( $q \geq 2^\ell$ , where  $\ell$  is the system's security parameter),  $q_S$  queries to the signing algorithm and  $q_V$  to the verifying*



algorithm in polynomial time  $t$ , then there exists an algorithm  $\mathcal{B}$  who can use  $\mathcal{A}_{II}$  to solve an instance of the GBDH problem with probability:  $Succ_{\mathcal{B}}^{GBDH} \geq Succ_{SDVPS, \mathcal{A}_{II}}^{EF-CMA} - \frac{q_V}{2^{\ell - q_H - q_S}}$  in the same time  $t$ .

*Proof:* Our overall strategy for the proof is as follows. We shall define a sequence  $\text{Game}_0, \text{Game}_1, \text{Game}_2, \text{Game}_3, \text{Game}_4$  of attack games. Each game operates on the same underlying probability space, in particular, the system’s parameter, public keys of the original signer Alice, the proxy signer Bob, the receiver Cindy and the values of the random oracle  $\mathcal{H}$ . We will prove that if there exists  $\mathcal{A}_{II}$  who can forge a valid signature of our SDVPS scheme, then there exists  $\mathcal{B}$  who can use  $\mathcal{A}_{II}$  to solve an instance of Gap Bilinear Diffie-Hellman problem. That is given a random instance  $(P, aP, bP, cP)$ ,  $\mathcal{B}$  can use  $\mathcal{A}_{II}$  to obtain the value of  $e(P, P)^{abc}$  with the help of Decisional Bilinear Diffie-Hellman (DBDH) Oracle.

$\mathcal{B}$  will simulate all the oracles in the proof. In the simulation,  $\mathcal{B}$  will maintain a list which is called *H-List* to record the hash queries and the corresponding values. We assume that  $\mathcal{A}_{II}$  is well-behaved in the sense that  $\mathcal{A}_{II}$  will never repeat the same queries in the simulation.

- **Game<sub>0</sub>.** We consider a Type II *EF-CMA* adversary  $\mathcal{A}_{II}$  with the success probability  $Succ_{SDVPS, \mathcal{A}_{II}}^{EF-CMA}$ . The original signer, Alice, selects his secret key  $x_A \in \mathbb{Z}_q^*$  and sets his public key as  $P_A = x_AP$ . The proxy signer Bob and designated verifier Cindy also generate their own secret/public key pairs  $(x_B, P_B)$  and  $(x_C, P_C)$ . Bob also publishes his identity  $ID_B$ .

The adversary  $\mathcal{A}_{II}$ , fed with  $(P_A, P_B, P_C)$  and  $x_B$ , can query the hash oracle  $H$ , the signing algorithm and the verify algorithm, and outputs  $(m^*, \sigma^*)$ , such that  $\text{Verify}(P_A, P_B, ID_B, x_C, m^*, \sigma^*) = \text{True}$ .

Let  $q_H, q_S, q_V$  denote the numbers of queries to the  $H$ , signing algorithm and verifying algorithm. The requirement is that  $m^*$  cannot be queried to the signing algorithm.

In any  $\text{Game}_i$ , we denote by **Forge**; the event  $\text{Verify}(P_A, P_B, ID_B, x_C, m, \sigma) = \text{True}$ . By definition, we have  $\Pr[\text{Forge}_0] = Succ_{SDVPS, \mathcal{A}_{II}}^{EF-CMA}$ .

- **Game<sub>1</sub>.** In this game,  $\mathcal{B}$  sets  $P_A = aP, Q_B = bP$  and  $P_C = cP$  where  $aP, bP, cP$  are the random instance of the Gap Bilinear Diffie-Hellman problem.  $\mathcal{B}$  also chooses  $b' \in \mathbb{Z}_q^*$  and sets  $P_B = b'P$ . Then  $\mathcal{B}$  returns  $(P_A, P_B, P_C, Q_B, b')$  to  $\mathcal{A}_{II}$ . Since  $a, b, c, b'$  are randomly chosen, therefore  $\Pr[\text{Forge}_1] = \Pr[\text{Forge}_0]$
- **Game<sub>2</sub>.** In this game,  $\mathcal{B}$  will simulate the random oracle  $H$ . There is a list *H-List* which maintains all the queries and answers consists of tuple  $(m_i, r_i, \sigma_i, \text{coin}_i)$ . Here  $(m_i, r_i)$  is the input of the  $H$  and  $\sigma_i$  is the output of  $H$ .  $\text{coin}_i = 1$  if  $r_i \cdot e(-P_C, Q_B)^{b'} = e(P, P)^{abc}$  and  $\text{coin}_i = 0$  otherwise. For any query  $(m_i, r_i)$  to the oracle  $H$ ,  $\mathcal{B}$  submits  $(r_i \cdot e(-P_C, Q_B)^{b'}, aP, bP, cP)$  to the DBDH oracle and DBDH oracle will tell  $\mathcal{B}$  whether  $r_i \cdot e(-P_C, Q_B)^{b'} = e(P, P)^{abc}$  or not
  1. If  $r_i \cdot e(-P_C, Q_B)^{b'} = e(P, P)^{abc}$ ,  $\mathcal{B}$  sets  $\text{coin}_i = 1$  and checks the *H-List*
    - (a) If there exists an item  $(m_i, \perp, \sigma_i, 1)$  in the *H-List*,  $\mathcal{B}$  returns  $\sigma_i$  as the answer.

- (b) Otherwise,  $\mathcal{B}$  chooses  $\sigma_i \in_R \mathbb{Z}_q^*$  such that there is no item  $(\cdot, \cdot, \sigma_i, \cdot)$  in the  $H\text{-List}$ .  $\mathcal{B}$  then adds  $(m_i, r_i, \sigma_i, 1)$  into the  $H\text{-List}$  and returns  $\sigma_i$  as the answer.
- 2. If  $r_i \cdot e(-P_C, Q_B)^{b'} \neq e(P, P)^{abc}$ ,  $\mathcal{B}$  chooses  $\sigma_i \in_R \mathbb{Z}_q^*$  such that there is no item  $(\cdot, \cdot, \sigma_i, \cdot)$  in the  $H\text{-List}$ .  $\mathcal{B}$  then adds  $(m_i, r_i, \sigma_i, 0)$  into the  $H\text{-List}$  and returns  $\sigma_i$  as the answer.

In the random oracle model, this game is clearly identical to the previous one. Hence  $Pr[\text{Forge}_2] = Pr[\text{Forge}_1]$ .

– **Game<sub>3</sub>**. In this game,  $\mathcal{B}$  simulates the signing algorithm. After receiving  $\mathcal{A}_{II}$ 's choice of the message  $m_i$ ,  $\mathcal{B}$  performs:

- 1. If there is a triple  $(m_i, r_i, \sigma_i, 1)$  in the  $H\text{-List}$ ,  $\mathcal{B}$  outputs  $\sigma_i$  as the signature.
- 2. Else  $\mathcal{B}$  chooses  $\sigma_i \in_R \mathbb{Z}_q^*$  such that there is no item  $(\cdot, \cdot, \sigma_i, \cdot)$  in the  $H\text{-List}$ . Then  $\mathcal{B}$  adds  $(m_i, \perp, \sigma_i, 1)$  to the  $H\text{-List}$  and outputs  $\sigma_i$  as the answer.

Then  $\mathcal{A}_{II}$  gets the value  $\sigma_i$  as the signature of  $m_i$ . Of course, this oracle simulates the signature perfectly, so  $Pr[\text{Forge}_3] = Pr[\text{Forge}_2]$ .

– **Game<sub>4</sub>**. In this game,  $\mathcal{B}$  simulates the verifying algorithm. After receiving  $\mathcal{A}_{II}$ 's request of  $(m_i, \sigma_i)$ ,  $\mathcal{B}$  checks :

- 1. If there is no item  $(\cdot, \cdot, \sigma_i, \cdot)$  in the  $H\text{-List}$ ,  $\mathcal{B}$  rejects  $(m_i, \sigma_i)$  as an invalid signature.
- 2. Else, there is an item  $(\cdot, \cdot, \sigma_i, \cdot)$  in the  $H\text{-List}$ :
  - (a) If this item has the form of  $(m_i, \perp, \sigma_i, 1)$  or  $(m_i, r_i, \sigma_i, 1)$ ,  $\mathcal{B}$  will accept it as a valid signature.
  - (b) Otherwise,  $\mathcal{B}$  rejects it as an invalid signature.

This makes a difference only if  $(m_i, \sigma_i)$  is a valid signature, while  $\sigma_i$  is not queried from the oracle  $H$ . Since,  $H$  is uniformly distributed, this case happens with probability less than  $\frac{1}{2^{\ell-q_H-q_S}}$ . Summing up for all verifying queries, we get  $Pr[\text{Forge}_3] - Pr[\text{Forge}_4] \leq \frac{q_V}{2^{\ell-q_H-q_S}}$ .

After **Game<sub>4</sub>** terminates,  $\mathcal{A}_{II}$  outputs a valid signature  $(m^*, \sigma^*)$  such that

$$\text{Verify}(P_A, P_B, ID_B, x_C, m^*, \sigma^*) = \text{True}.$$

That is, there is an item  $(\cdot, \cdot, \sigma^*, \cdot)$  in the  $H\text{-List}$ . By the definition of the EF-CMA adversary model,  $m^*$  can not be queried in the sign oracle, so  $\sigma^*$  is returned as the hash value of  $\mathcal{A}'_{II}$ 's query  $(m^*, r^*)$ . That is to say  $(m^*, r^*, \sigma^*, 1)$  is in the  $H\text{-List}$  and  $r^* \cdot e(P_C, -Q_B)^{b'} = e(P, P)^{abc}$ . Note that  $P_C = cP, Q_B = bP$  and  $b'$  is randomly chosen by  $\mathcal{B}$ , so  $\mathcal{B}$  can compute  $e(P, P)^{abc} = r^* \cdot e(bP, -cP)^{b'}$ . Therefore, given  $aP, bP, cP$ ,  $\mathcal{B}$  successfully solves an instance of the GBDH problem with probability:  $Succ_{\mathcal{B}}^{GBDH} \geq Succ_{SDVPS, \mathcal{A}_{II}}^{EF-CMA} - \frac{q_V}{2^{\ell-q_H-q_S}}$ . ■

**Theorem 3.** *If the Type III Adversary  $\mathcal{A}_{III}$  (that is the original signer Alice) can forge a valid signature of the proposed scheme with success probability  $Succ_{SDVPS, \mathcal{A}_{III}}^{EF-CMA}$  after making  $q_H$  queries to the  $H_1 : \{0, 1\}^* \rightarrow \mathbb{Z}_q^*$  ( $q \geq 2^\ell$ ,  $\ell$  is the system's security parameter),  $q_S$  queries to the signing algorithm and  $q_V$  to the verifying algorithm in some polynomial time  $t$ , then there exists an algorithm  $\mathcal{B}$  who can use  $\mathcal{A}_{III}$  to solve an instance of the GBDH problem with probability:  $Succ_{\mathcal{B}}^{GBDH} \geq Succ_{SDVPS, \mathcal{A}_{III}}^{EF-CMA} - \frac{q_V}{2^{\ell-q_H-q_S}}$  in the same time  $t$ .*

*Proof.* The whole proof is almost the same as the above, except that Given  $aP, bP, cP, \mathcal{B}$  sends  $(P_A = a'P, P_B = aP, Q_B = bP, P_C = cP, a')$  to this Type III adversary.

At last,  $\mathcal{A}_{III}$  outputs a valid signature  $(m^*, \sigma^*)$  such that  $\text{Verify}(m^*, \sigma^*, P_A, P_B, Q_B, c) = \text{True}$ . That is to say  $(m^*, r^*, \sigma^*, 1)$  is also in the  $H - \text{Liast}$ . Since  $\sigma^*$  is a valid signature of the message  $m^*$ , then  $r^* \cdot e(P_C, -Q_B)^{a'} = e(P, P)^{abc}$ . Note that  $P_C = cP, Q_B = bP$  and  $a'$  is randomly chosen by  $\mathcal{B}$ , so  $\mathcal{B}$  can compute  $e(P, P)^{abc} = r^* \cdot e(bP, -cP)^{a'}$ . Therefore, given  $aP, bP, cP, \mathcal{B}$  successfully solves an instance of the GBDH problem with probability:  $\text{Succ}_{\mathcal{B}}^{\text{GBDH}} \geq \text{Succ}_{\text{SDVPS}, \mathcal{A}_{III}}^{\text{EF-CMA}} - \frac{qv}{2^{\ell - q_H - q_S}}$ . ■

## 6 Comparison

In this section, we compare the signature length of our short designated verifier signature scheme (*SDVPS*) with Wang’s scheme in [12]. The signature of Wang’s scheme is  $(r_p, K, D, s)$  where  $r_p, K, D \in \mathbb{Z}_p$  and  $s \in \mathbb{Z}_q$ . Let  $|\mathbb{Z}_p|$  denote the bit length of the element in  $\mathbb{Z}_p$  and  $|\mathbb{Z}_q|$  denote the the bit length of the element in  $\mathbb{Z}_q$ , we have the following table.

Scheme	Signature Length	$p : 1024; q : 160$
Wang’s Scheme	$3 \mathbb{Z}_p  +  \mathbb{Z}_q $	3232 bits
Our Scheme	$ \mathbb{Z}_q $	160 bits

One can find that the signature length of our *SDVPS* scheme is *dramatically* decreased, which is more applicable in the networks with limited bandwidth. One can also find that the implementation of out scheme needs the *bilinear pairing*, how to get a *SDVPS* scheme without the need of pairing is an open problem.

## 7 Conclusion

We have presented a new designated verifier proxy signature scheme, which we believe is the shortest among all the known designated verifier proxy signatures. We prove that our scheme offers transcript simulation as a normal designated signature. We also prove that our scheme is secure under random oracle model.

## References

1. D. Chaum. Zero-knowledge undeniable signatures. In *Advances in Cryptology, Proc. EUROCRYPT 1991*, LNCS 547, pages 458–464. Springer–Verlag, Berlin, 1991.
2. J. Z. Dai, X. H. Yang, and J. X. Dong. Designated-receiver proxy signature scheme for electronic commerce. In *Proc. of IEEE International Conference on Systems, Man and Cybernetics*, pages 384–389. IEEE Press, 2003.

3. S. Galbraith and W. Mao. Invisibility and anonymity of undeniable and confirmer signatures. In *Proc. of CT-RSA 2003*, LNCS 2612, pages 80–97. Springer–Verlag, Berlin, 2003.
4. M. Jakobsson, K.Sako, and R. Impagliazzo. Designated verifier proofs and their applications. In *Advances in Cryptology, Proc. EUROCRYPT 1996*, LNCS 1070, pages 143–154. Springer–Verlag, Berlin, 1996.
5. F. Laguillaumie and D. Vergnaud. Designated verifiers signature: Anonymity and efficient construction from any bilinear map. In *Fourth Conference on Security in Communication Networks '04 (SCN 2004)*, LNCS 3352, pages 107–121. Springer–Verlag, Berlin, 2004.
6. B. Libert and J.-J. Quisquater. Identity based undeniable signatures. In *Proc. of CT-RSA 2004*, LNCS 2964, pages 112–125. Springer–Verlag, Berlin, 2004.
7. M. Mambo, K. Usuda, and E. Okamoto. Proxy signatures for delegating signing operation. In *Proc. of the Third ACM Conf. on Computer and Communications Security*, pages 48–57, 1996.
8. S. Saeednia, S. Kramer, and O. Markovitch. An efficient strong designated verifier signature scheme. In *The 6th International Conference on Information Security and Cryptology (ICISC 2003)*, LNCS 2971, pages 40–54. Springer–Verlag, Berlin, 2003.
9. R. Steinfeld, H. W. L. Bull, and J. Pieprzyk. Universal designated-verifier signatures. In *Advances in Cryptology–ASIACRYPT 2003*, LNCS 2893, pages 523–543. Springer–Verlag, Berlin, 2003.
10. R. Steinfeld, H. W. L. Bull, and J. Pieprzyk. Efficient extension of standard schnorr/rsa signatures into universal designated-verifier signatures. In *Public Key Cryptography, Proc. PKC 2004*, LNCS 2947, pages 86–100. Springer–Verlag, Berlin, 2004.
11. W. Susilo, F. Zhang, and Y. Mu. Identity-based strong designated verifier signature schemes. In *Proceedings of the Information Security and Privacy, 9th Australasian Conference (ACISP 2004)*, LNCS 3108, pages 313–324. Springer–Verlag, Berlin, 2004.
12. G. Wang. Designated-verifier proxy signatures for e-commerce. In *the IEEE 2004 International Conference on Multimedia and Expo (ICME 2004)*, pages 1731–1734. IEEE Press, 2004.
13. H. Wang and J. Pieprzyk. Efficient one-time proxy signature. In *Advances in Cryptology–Aisacrypt 2003*, LNCS 2894, pages 507–522. Springer–Verlag, Berlin, 2003.
14. F. Zhang and K. Kim. Id-based blind signature and proxy signature from bilinear pairings. In *In: Information Security and Privacy (ACISP 2003)*, LNCS 2727, pages 312–323. Springer–Verlag, Berlin, 2003.
15. K. Zhang. Threshold proxy signature schemes. In *In Proc. Information Security (ISW 1997)*, LNCS 1396, pages 282–290. Springer–Verlag, Berlin, 1997.

# An Embedded Gateway Based on Real-Time Database

Zhiping Jia<sup>1</sup> and Xinxiao Qiao<sup>2</sup>

<sup>1</sup> School of Computer Science and Technology, Shandong University,  
Jinan, 250061, P.R. China  
zhipingj@sdu.edu.cn

<sup>2</sup> Department Of Computer Science and Technology, Shandong Institute Light Industry,  
Jinan, 250018, P.R. China  
qxxyyn@126.com

**Abstract.** Because of the limitation of the traditional gateway in the distributed monitoring system, the access of the protocol translation gateway will increase drastically with the massive usage of Ethernet. By using M/M/1 queuing theories, we present a framework of gateway based on the real-time database. The whole gateway is divided into two parts. One is linked to the monitoring subnet to collect the data with a polling method and deposit them in real-time database of the gateway. The other part is used for accepting the TCP connecting request, accessing the real-time database of the gateway. By analysis of the M/G/1 theory, it can be concluded that this framework can not only update the data more quickly in the distributed observing and controlling subnet, but also reduce the rejection of connections and loss of packet caused by the full queue buffer effectively when the number of TCP links become larger.

## 1 Introduction

Ethernet has already got extensive application in the distributed monitoring system because of its characters such as well-opening, extensive using and cheaper. The traditional monitoring systems mostly adopt Ethernet in the information layer, and generally use different field buses in the controlling layer and equipment layer. However, with the perfection of Ethernet and the development of embedded technology, the embedded network technology is a very fashionable topic for discussion. Ethernet has already permeated through the controlling layer and equipment layer of the monitoring system, which makes the access of the protocol translation gateway increase rapidly. And thus the limitation of the traditional gateway shows at this moment (Fig.1 shows the overall picture of network interconnection of a typical distributed monitoring system).

In this paper, we will reconstruct the protocol translation gateway for the distributed monitoring field, and present a framework which resolves the limitation existed in the traditional gateway when there are too many TCP connections. The framework can improve the network throughput and enhance the characterizes of real-time and security of the distributed monitoring system.

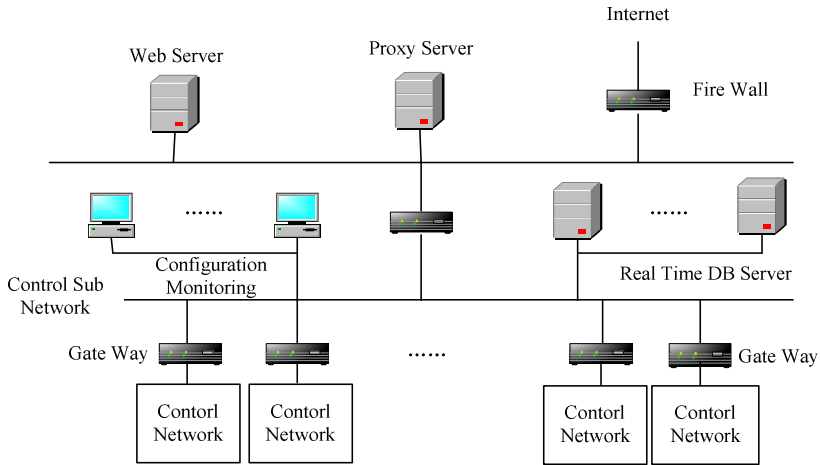


Fig. 1. The Connection Structure of Distributed Monitoring Network

## 2 Analysis of the Framework

The protocol translation gateway is the contractual media between Ethernet and the monitoring subnet. Ethernet protocol can carry on multi-connections and complete the data transmission of multi-connections at one time. However, what the monitoring subnet uses is the serial chain, which can carry on only one connection at a certain moment. And only one connection's data transmission is allowed. When multi-connections are needed, the connection requests should queue and wait. Nowadays, most protocol translation gateway carries on data transmission in way of principal-and-subordinate ask-and-answer in the monitoring field. Thus, it will cause the block problem of connections when TCP connections are more. The larger the number of connections is, the slower the response time of the whole system is.

The framework of the traditional gateway is shown as Fig.2. When the serial chain is busy, the connect request can only wait. With the popularization of Ethernet in the industry automatic field, the access of the gateway will increase doubly, and then the response time of the whole system will increase, as a result, it will influence the real-time character of the whole system.

## 3 Model Analysis of the Framework

It can be known from the queuing theory [1] [2] [3] that the gateway model shown in Fig. 2 is a classics queue model -M/M/1[4]. In Fig.3, in the moment  $t+\Delta t$ , we set the number of customers who are accepted by system and queuing in the window is  $x$ . And we called it state  $X$ , whose probability is  $P_x(t+\Delta t)$ .

Here, in the block( $t,t+\Delta t$ ):

(1)The probability of one customer reaching is  $\lambda\Delta t+O(\Delta t^2)$ ,and the probability of no customer reaching is  $1-\lambda\Delta t+O(\Delta t^2)$ .

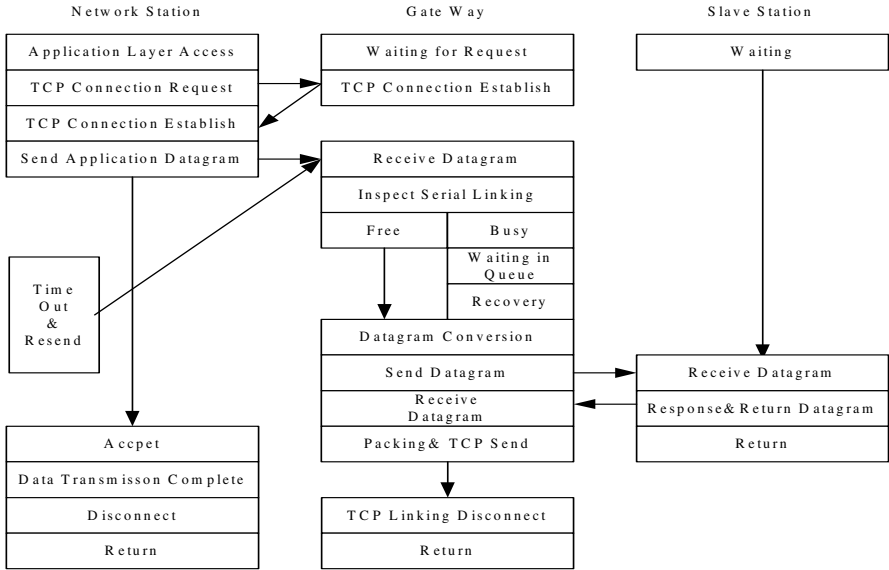


Fig. 2. Framework of the traditional gateway

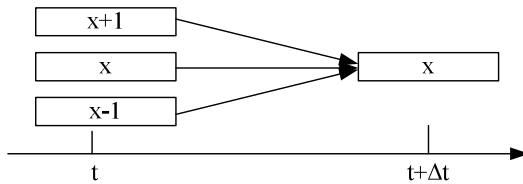


Fig. 3. The transition diagram of state X

(2)When there is customer receiving service, the probability of one customer leaving away is  $\mu\Delta t + O(\Delta t^2)$ ; and the probability of no customer leaving away is  $1 - \mu\Delta t + O(\Delta t^2)$ .

(3)The probability of more than one customer reaching or leaving away is  $O(\Delta t^2)$ , which can be neglected.

According to the whole probability theorem, the probability of state X is:

$$P_0(t+\Delta t) = P_0(t)(1-\lambda\Delta t) + P_1(t)(1-\lambda\Delta t)\mu\Delta t$$

$$\frac{dP_x(t)}{dt} = P_{x+1}(t) + P_{x-1}(t) - (\lambda + \mu)P_x(t)$$

We set  $\Delta t \rightarrow 0$ , and then have a differential equation:

$$\frac{P_x(t + \Delta t) - P_x(t)}{\Delta t} = P_{x+1}(t)\mu + P_{x-1}(t)\lambda - (\lambda + \mu)P_x(t) + \frac{O(\Delta t^2)}{\Delta t}$$

$$\frac{dP_0(t)}{dt} = P_0(t)\mu - \lambda P_1(t)$$

When the serving rate  $\mu$  is greater than the reaching rate  $\lambda$  and  $t \rightarrow \infty$  is stable, the state probability has nothing to do with time, namely, the differential item of differential equation is zero. Then the probability of state  $X$  is :

$$P_0=1-\rho$$

$$P_x=(1-\rho)\rho^x \quad x \geq 1$$

Here :  $\rho = \lambda / \mu$

Using the system state probability, we can have the relevant operation index of the system:

(1) The average number of customers in the system is:

$$L_s = \sum_{x=0}^{\infty} xP_x = \sum_{x=0}^{\infty} x(1-\rho)\rho^x = \frac{\rho}{1-\rho}$$

(2) The average number of customers waiting in the queue is:

$$L_q = \sum_{x=0}^{\infty} (x-1)P_x = \sum_{x=0}^{\infty} xP_x - \sum_{x=0}^{\infty} P_x = L_s - \rho = \frac{\rho^2}{1-\rho} = \frac{\rho\lambda}{\mu-\lambda}$$

(3) The expected value of the time customers waiting in the queue is:

$$W_q = L_q / \lambda = \frac{\rho}{\mu-\lambda}$$

(4) The expected value of the time customers waiting and being serviced is:

$$W_c = W_q + 1/\mu = \frac{1}{\mu-\lambda}$$

Because we use the serial circuit to accept the data of subnet communication unit, the communication speed of the serial circuit will cause greater influence on the serving rate.

The time of one service includes: the affirmation time of a connect request, the conversion time of datagram, the transmission delay of message reaching the subnet communication unit, the treatment time of the subnet communication unit, the transmission delay of message in the communication unit and the time of sending the TCP datagram.

If we think that the bandwidth of the upper network of the gateway is above 10M and the communication amount is general, then we can ignore the time of TCP connecting and transmitting delay. If we use the constant-length datagram to be transmitted, we set the datagram length is 20Byte. And we use an asynchronous communication way that has 8 data bits, one beginning bit and one ending bit. When the serial communication speed is between 600bps to 1 Mbps, we separately get the corresponding two-way transmission time, the approximate service rate, and the reaching rate, which are shown in Table 1.

It can be found out from Table 1, that with the serial chain speed increasing, the delay of datagram transmit in the serial chain is reduced, and when the speed is quickly enough, we can't ignore the time of TCP connecting and of transmit. Especially in a situation when the speed of the serial chain is slower and the TCP connections are more, the congestion of requests of the monitoring network will be caused, which slows down the speed of the data reading.



**Table 1.** Serial link transmit speed & delay and restriction of average arriving rate

Transmit Speed ( kbps)	0.6	1.2	4.8	9.6	19.2	56	1000
Datagram Length (bit)	240	240	240	240	240	240	240
Transmit Delay ( ms)	400	200	50	25	12.5	6.25	0.24
Restriction of Average Arriving Rate	0.75	1.5	6	12	24	70	250

The model analysis mentioned above is got under a condition that the queue buffer is infinite. But in the real situation, because of the hardware restraint, the queue quantity allowed is not too large. So when TCP connections are more, it will cause the problem of rejection of connections and loss of datagram, and influences the security of the whole system.

## 4 Improvement of Framework

The serial link has single access characteristic in traditional gateway, that is, only one connection can access at a certain moment, while others need to queue, which causes data access time to become longer. Thus it tends to cause requests congestion, connections timeout and datagram loss when the TCP connections are more, so we provide a new gateway framework based on real-time database. Profited from the swift development of embedded system, this framework can effectively operate and query data and process massive capacity data storage.

Designed a small-scale real-time database in gateway, the whole gateway can be divided into two parts. One part is connected with observing and controlling subnet, collecting data of subnet by a poll method and then storing them into the real-time database in gateway. The other part is a network workstation, which accesses the real-time database in gateway and retrieves data wanted directly, thus avoiding problems of queuing TCP connections caused by monopolization of serial link. The bridge between two parts is real-time database.

Under this way, we set the framework between the gateway and network access workstation to be Client/Server pattern, where the gateway is a database server and network access workstation is client. Gateway application is in state of waiting for connection and responsible to provide data needed, format transformation of datagram, address storage and send, etc. As to control subnet, gateway is thought to be the main site, which retrieves data by polling subsidiary site according to information of real-time database. Fig.4 shows this connection access process.

For example, we adopt MODBUS protocol[5]. Network workstation accepts TCP connection request, establishes connection, and is ready to receive datagram. After that, it parses datagram according to MODBUS/TCP protocol and gets MODBUS protocol frame. Then it parses MODBUS protocol frame, gets request command sent by PC, and queries real-time database according to request command, sends back the data encapsulated by MODBUS and MODBUS/TCP protocol to request client, and then

waits for request of TCP link removal. When it receives removal request, it will remove TCP link.

The other part retrieves commands from database by way of the access pattern of subnet controlling, sends MODBUS frame encapsulated according to MODBUS protocol to serial link, and then waits for return of data. After accepted data, it will parse MODBUS frame and write these data into real-time database, then retrieve the next command, and repeat the process above. As a result, all the data of real-time database update instantly.

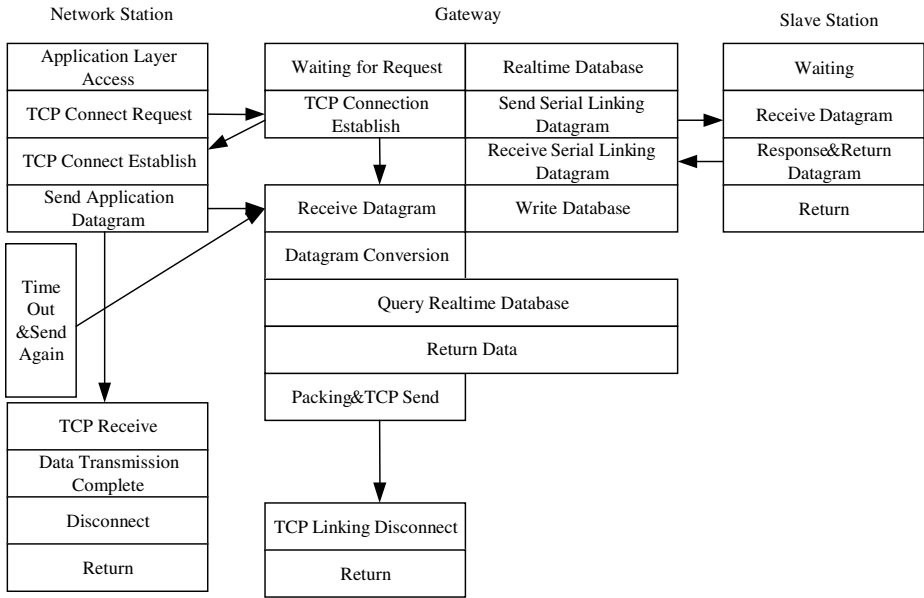


Fig. 4. The improved framework of gateway

## 5 Analysis of Gateway Framework Model with Database

### 5.1 Analysis of Control Subset Model

In the transfer mode of control subnet, the main communication source is gateway, which repeatedly reads data from subsidiary site and stores them in the database of gateway.

The above pattern is M/G/1 model [6][7]. We suppose the probability distributing of n is F(n), the probability mother function of F(n) is G(Z), the mean value is E(n), the variance is V(n), and the probability of accessing node n is P(n).

We suppose access time of every node is  $t_i$  ( $i=1,2,\dots,n$ ), so total service time is:

$$T_n = \sum_{i=0}^{\infty} xP_x$$

After all calculation, the last result is as follows:

$$\begin{aligned}
E(T_n) &= G'(1)H'(0) = E(n)E(t_i) \\
V(T_n) &= T_n''(0) - [T_k'(0)]^2 \\
&= G'(1)H''(0) + G''(1)[H'(0)]^2 - [G'(1)H''(0)]^2 \\
&= E(n)V(t_i) + [E(t_i)]^2V(n)
\end{aligned}$$

The average waiting time is as follows:

$$W_q = \frac{Lq}{\lambda} = \frac{\lambda^2 \rho^2 + (\lambda/\mu)^2}{2\lambda(1-\lambda/\mu)} = \frac{\lambda[V(T_n)]^2 + [\lambda E(n)E(t_i)]^2}{2\lambda(1-\lambda E(n)E(t_i))}$$

From above calculation, we can see that waiting time of client is relative to how much data needed to read in and the speed of serial link. The less data needed to be read in and the faster the serial link is, the swifter of data update speed is and the more real-time will be. In this solution, if the accessing chance of every data is equal, the existing time while the data be accessed is half of client waiting time.

As the operation of database avoiding the single access of serial link, it can improve the throughput of gateway and effectively avoid problem of request congestion.

## 5.2 Analysis of TCP Connection Response

The time of forth-and-back (RTT) of every TCP connection will be shorten because the real-time database is introduced. There is only one connection that can transfer data in serial link at one time in traditional gateway, so the TCP congestions do not exist. By introducing real-time database, it may generate congestion problem for the fast query time in database. As many papers have discussed TCP congestion problem, we will not go deep into these discuss. We just suppose the number of the TCP connection is not large enough to cause the TCP congestion. Thus as to the serial links, the TCP transfer time can be ignore.

As several TCP connections could be established, it will not have problems of TCP rejection of connection and loss of datagram, and this improves the security of system.

## 5.3 Analysis of the Whole Data

We suppose there are 10 control points, every control point averagely access 8 bytes and return 8 bytes (decided by serial link protocol), and the communication speed of control subnet is 19200baud. When the number of TCP connections change, we will get the response time of gateway (Shown in Table.2). While we set 10 TCP connections, we get response time of gateway (Shown in Table.3) when the connection speed of control subnet changes.

According to contrast, we can observe that the improved gateway framework is worse in real-time than traditional framework when there is only one TCP connection but more points in control subnet. But this framework will be better in real-time than traditional framework while the number of TCP connections increases and there are less points in control subnet. Especially, when the number of TCP connections increases, the improvement of real-time character will be in evidence. Indeed, real-time character of gateway is also related to the speed of control subnet.

**Table 2.** Number of TCP connection and response time

TCP Link Number	Response Time of Source Gateway (ms)	Response Time of New Gateway (ms)	
		Return Time of Gateway(ms)	Return Time of Control Subnet (ms)
1	12	<1	60
2	24	<1	60
5	60	<1	60
6	75	1	60
8	115	1	60
10	155	2	60
11	180	3	60
13	245	3	60

**Table 3.** Speed and response time of control subnet

Control Subnet Speed	Response Time of Source Gateway(ms)	Response Time of New Gateway(ms)	
		Return Time of Gateway (ms)	Return Time of Control Subnet (ms)
1200	785	2	375
2400	445	2	205
4800	275	2	120
9600	195	2	80
19200	155	2	60
56000	125	2	45

## 6 Conclusion

From above analysis, we can infer that by means of real-time database, we will effectively solve the problems caused by multiple TCP connections in distributed monitoring field, which include queue request, rejections of connection and loss of control network protocol packets, and improve properties of real-time and security of system. For the gateway mode based on real-time database, we also need some improvements on the following aspects:

(1) Compatibility problem with the traditional gateway. As both gateway have separate advantages and disadvantages, and real-time character is strong or weak under different scenarios, so we should work on these compatibility problems in future models.

(2) Construction problem of the real-time database. In distributed monitoring field, if the real-time property of some variables is inadequate, the process such as alarm and remote control will induce fatal problems. But other variables change slowly, such as pressure and flux. We should establish different hierarchical databases according to varied characteristic of variables.

(3) Capacity problem of the database. According to the above analysis, the more control points in database, the longer each poll time will be. And it will go against the improvement of real-time. How to control the number of points, that is, the capacity of database, to make best the real-time character of the whole system, is also problems needed to be solved in future.

## References

1. N.U.Prabhu: A Bibliography Of Books And Survey Papers On Queueing Systems—Theory And Applications, Queueing Systems 2,393-398,1987
2. Ivan Atencia, Pilar Moreno: A Discrete-Time Geo/G/1 Retrial Queue with General Retrial Times, Queueing Systems 48, 5–21, 2004
3. Muckai K. Girish , Jian-Qiang Hu: An Interpolation Approximation For the GI/G/1 QueueBased on Multipoint Padé Approximation Queueing Systems 26,269–284,1997
4. Nikhil Bansal: On the Average Sojourn Time Under M/M/1/SRPT, Operations Research Letters 33,195–200,2005
5. Modbus Messaging On TCP/IP Implementation Guide(Rev 1.0), Modbus.org, 2002
6. Yong J. Kima, Michael V. Manninob: Optimal Incentive-compatible Pricing for M/G/1 Queues, Operations Research Letters 31,459–461,2003
7. Krishna Kumar B., Pavai Madheswari S.: M/sup x//G/1 Retrial Queue With Multiple Vacations and Starting Failures, Opsearch vol.40,no.2,115-37, 2003

# Efficient Authentication Scheme for Routing in Mobile Ad Hoc Networks

Shidi Xu, Yi Mu, and Willy Susilo

School of Information Technology and Computer Science,  
University of Wollongong, Australia  
{sdx86, wsusilo, ymu}@uow.edu.au

**Abstract.** The security deployment in mobile ad hoc networks is frequently hampered by resource constraints. The current routing systems of mobile ad hoc networks deploy very weak security techniques in order to copy with the computational overhead and bandwidth consumption. In this paper, we present an ID-based online/offline signature scheme which provides a full scale of security with sound performance. We show that our scheme is secure against existential forgery under adaptive chosen message attacks.

**Keywords:** Authentication, Digital Signature, MANET, Routing, AODV.

## 1 Introduction

The security technology deployed in the existing mobile ad hoc networks (MANET) is very weak, since the resource constraint makes complex security computations infeasible. Several well-known MANET routing protocols such as DSR [7] and AODV [8] were designed without a security consideration. Consequently, MANET routing systems face a number of security threats, from basic spoofing attacks to more complex rushing attacks. How to provide full-scale security to MANET with a low computational overhead and bandwidth consumption becomes an open problem to security researchers.

The security deployment to MANET is stunted by cryptographic techniques themselves, since they are expensive to configure and perform. In a MANET, each node is highly mobile, and hence it requires the routing operations to be accomplished within their lifetime; otherwise the routing information will not be able to represent the current topology condition. In addition to computational overhead, MANET also has problems in key distribution. This is particularly important in routing, because in a routing system, mobile nodes are not aware of other nodes that are out of their radio signal broadcasting diameter. This is usually roughly handled by a pre-key distribution phase. However, in an ad hoc network, which is formed impromptu, the authentication between nodes is performed in a cursory manner.

**Our Contribution.** In this paper, we introduce a novel authentication scheme to tackle the problem of computational overheads in MANET. We devise a novel

digital signature scheme that is especially feasible for authentication in MANET. In our scheme, a signing operation is split into two phases: offline phase and on-line phase. The major computational overhead is shifted to the offline phase, whereas the online phase requires only a very low computation overhead to achieve a full scale of authentication. Moreover, the public key distribution problem is solved by using node's identity such as IP or MAC address as its public key. We will also describe how this signature can be used for securing an AODV routing system.

**Organization of the paper.** The rest of the paper is organized as follow. In section 2, we introduce several secure routing protocols, the concept of online/offline signature, and review previous works. In section 3, we give some preliminaries of bilinear pairings and definitions. In section 4, we define the generic scheme and attack model. In section 5, we present our scheme and analyze its security. In section 6, we describe how to apply our scheme to the AODV routing system. In the last section, we conclude the paper.

## 2 Previous Work

### 2.1 Secure Routing Protocol

To protect MANET routing systems against various attacks, a sound authentication scheme must be deployed. There have been several schemes in the literature. Each of them uses a different method in providing sender authentication and message integrity.

Ariadne, proposed by Hu, Perrig and Johnson [6], is a secure on-demand routing protocol based on DSR. The security of Ariadne generally relies only on highly efficient symmetric-key cryptography. It assumes a pre-deployed secret shared between the sending node and targeting node. The authentication between intermediate nodes is done using the TESLA authentication protocol. During the transmission of route requests, each intermediate node appends a MAC generated using TESLA key. This MAC will be authenticated when a route reply is transmitted back to the originator. Since the TESLA authentication protocol is used, each node must be loosely time synchronized to decide the validity of TESLA keys, which becomes the major drawback of Ariadne.

SAODV [10] is a security extension of the AODV routing protocol. Since the routing operation in AODV is very simple, its security requirement can be easily satisfied. SAODV uses conventional digital signatures to protect routing messages. However, it neither deploys the public key certificate nor assumes pre-shared secret between nodes. Each sending node signs its own public key along with routing messages. The key distribution problem is loosely solved with some compromise of security.

### 2.2 Online/Offline Signature

The online/offline digital signature scheme was firstly introduced by Even, Goldreich and Micali [4]. The basic concept of their scheme is splitting the signature

generation algorithm into two phases: offline phase and online phase. To achieve efficient performance when a message is coming to be signed, they utilize an offline phase to handle the most costly computation. When a message is ready, the online phase can be performed extremely efficient to generate the required signature. On drawback of their scheme is that the size of public key and resulting signature is likewise very large since one-time signature is used.

Zhang, Mu and Susilo [11] proposed the first online/offline signcryption scheme from bilinear pairings. The online signing phase is also very efficient in their scheme, which requires approximately one hash. The size of the resulting signature is reduced to  $\log_2 p + \log_2 \rho + 160$ , where  $p$  is the order of cyclic additive group and  $\rho$  is the safe length of that group, where the underlying cryptographic assumption still holds. Although this scheme has its merit, it is out of the scope of our aim since we consider the efficient authentication only.

### 3 Bilinear Pairings

Let  $\mathbb{G}_1$  be a cyclic additive group generated by  $P$ , with a prime order  $q$ , and  $\mathbb{G}_2$  be a cyclic multiplicative group with the same prime order  $p$ . Let  $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$  be a map with with the following properties:

1. Bilinearity:  $e(aP, bQ) = e(P, Q)^{ab}$  for all  $P, Q \in \mathbb{G}_1, a, b \in \mathbb{Z}_q^*$ ;
2. Non-degeneracy: There exists  $P, Q \in \mathbb{G}_1$  such that  $e(P, Q) \neq 1$ ;
3. Computability: There is an efficient algorithm to compute  $e(P, Q)$  for all  $P, Q \in \mathbb{G}_1$ ;

The Non-degeneracy implies that when  $P$  is the generator of  $\mathbb{G}_1$ ,  $e(P, P)$  is the generator of  $\mathbb{G}_2$ . We call such bilinear map as an admissible bilinear pairing. Problems considered in the additive group  $\mathbb{G}_1$  are:

- **Decisional Diffie-Hellman Problem (DDHP):** For  $a, b, c \in \mathbb{Z}_q^*$ , given  $P, aP, bP, cP$  decide whether  $c \equiv ab \pmod{q}$ .
- **Computational Diffie-Hellman Problem (CDHP):** For  $a, b \in \mathbb{Z}_q^*$ , given  $P, aP, bP$  compute  $abP$ .

In bilinear pairings, Decision Diffie-Hellman problem (DDHP) is easy and Computational Diffie-Hellman problem (CDHP) is still hard. That is, for  $a, b \in \mathbb{Z}_q^*$ , given  $P, aP, bP$ , computing  $abP$  is infeasible.

**Definition 1.** A group  $\mathbb{G}$  is a gap Diffie-Hellman (GDH) if there exists a polynomial time probabilistic algorithm to compute the decisional Diffie-Hellman problem but exists no such algorithm to solve the computational Diffie-Hellman problem in  $\mathbb{G}$ .

Above system parameters can be obtain through running the **GDH Parameter Generator** [3]  $\mathcal{IG}$  which takes a security parameter  $k \in \mathbb{Z}^+$  as input, runs in polynomial time in  $k$ , and outputs a prime number  $q$ , the description of two groups  $\mathbb{G}_1, \mathbb{G}_2$  of order  $q$ , and the description of an admissible bilinear map  $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ .



**Definition 2.** *The advantage of an algorithm  $\mathcal{A}$  in solving CDHP in group  $\mathbb{G}$  is*

$$Adv_{\mathcal{A}}^{CDH} = \Pr[\mathcal{A}(P, aP, bP) = abP : a, b \stackrel{R}{\leftarrow} \mathbb{Z}_q^*]$$

where the probability is over the choice of  $a$  and  $b$ , and the coin tosses of  $\mathcal{A}$ . We say that an algorithm  $\mathcal{A}(t, \epsilon)$ -breaks CDHP in  $\mathbb{G}$  if  $\mathcal{A}$  runs in time at most  $t$ , and  $Adv_{\mathcal{A}}^{CDH} > \epsilon$ .

## 4 The Model

In this section we formalize the general online/offline digital signature paradigm. This paradigm is extended to elicit our ID-based scheme.

### 4.1 Generic Scheme

Online/offline digital signature scheme  $\mathcal{DS}$  is comprised of four polynomial time algorithms: *KeyGen*, *OffSign*, *OnSign*, and *Verify*, called *key generation algorithm*, *offline signing algorithm*, *online signing algorithm*, and *verification algorithm*, respectively. The first three algorithms are probabilistic.

**KeyGen.** On input  $1^k$ , the algorithm produces a pair of matching public and secret keys  $(pk, sk)$ .

**OffSign.** On input  $(sk, r)$ , where  $r$  a signing parameter, the algorithm returns an offline signature  $S$ .

**OnSign.** On input  $(S, m)$ , where  $S$  is the offline signature and  $m$  is the message, the algorithm returns an online signature  $\sigma$ .

**Verify.** On input  $(pk, m, S, \sigma)$ , the algorithm returns 1 (*accept*) or 0 (*reject*).

The security for signature schemes was defined by Golwasser, Malia and Rivest [5] as secure against existential forgery under adaptive chosen message attacks (EF-CMA). We extend this notion to online/offline signature schemes as follow:

**Definition 3. (Security)** [5] *The online/offline signature scheme*

$$\mathcal{S} = \langle \text{KeyGen}, \text{OffSign}, \text{OnSign}, \text{Verify} \rangle$$

*is existential unforgeable under adaptive chosen message attacks if it is infeasible for a forger to produce a valid message-signature pair after obtaining polynomially many signatures on a message of its choice from the signer.*

*Formally, for every probabilistic polynomial forger  $\mathcal{A}$  such that:*

$$Adv(\mathcal{A}) = \Pr \left[ \begin{array}{l} (pk, sk) \leftarrow \text{KeyGen}(1^k); \\ \text{for } i = 1, 2, \dots, k, r; \\ m_i \leftarrow \mathcal{A}(pk, m_1, S_1, \sigma_1, \dots, m_{i-1}, S_{i-1}, \sigma_k); \\ S_i \leftarrow \text{OffSign}(sk, r), \sigma_i \leftarrow \text{OnSign}(S_i, m); \\ (m, S, \sigma) \leftarrow \mathcal{A}(pk, m_1, S_1, \sigma_1, \dots, m_k, S_k, \sigma_k); \\ m \neq m_1, \dots, m_k \text{ and } \text{Verify}(pk, m, S, \sigma) = \text{accept}; \end{array} \right] \leq \epsilon$$

### 4.2 Attack Model

The formal attack model for ID-based signature scheme was firstly generalized by Cha and Cheon in [2], which is called *existential forgery under adaptive chosen message and ID attack*(EF-IOS-CMA).

We can define our game between an attacker  $\mathcal{A}$  and a challenger  $\mathcal{C}$  as follow:

1.  $\mathcal{C}$  runs **Setup** to obtain the system parameters which are given to  $\mathcal{A}$ .
2.  $\mathcal{A}$  runs message hash query, ID extraction query, online and offline signing query to obtain necessary information.
3.  $\mathcal{A}$  finally outputs  $(ID, m, S, \sigma)$ , where  $ID$  is an identity,  $m$  is a message,  $S$  is offline signature, and  $\sigma$  is online signature, such that  $ID$  and  $(ID, m)$  are not in the inputs to extraction query and signing query.  $\mathcal{A}$  wins the game if  $(ID, m, S, \sigma)$  is valid.

**Definition 4.** *The success probability of winning the above game is defined by  $Succ_{\mathcal{A}}^{EF-IOS-CMA}(\ell)$ . An online/offline signature scheme is secure if the success probability of the above attack is negligible. In other words,*

$$Succ_{\mathcal{A}}^{EF-IOS-CMA}(\ell) \leq \epsilon.$$

## 5 Our Scheme

Based on the general scheme, our ID-based online/offline scheme consists five algorithms: **Setup**, **Extract**, **OffSign**, **OnSign**, **Verify**.

**Setup.** Given  $\mathbb{G}_1$  and its generator  $P$ , pick a random  $s \in \mathbb{Z}_q^*$ , and set  $P_{pub} = sP$ .

Choose a cryptographic hash function  $H_0 : \{0, 1\}^* \rightarrow \mathbb{G}_1$  and  $H_1 : \{0, 1\}^* \times \mathbb{G}_1 \rightarrow \mathbb{Z}_q^*$ . The system parameters are  $(P, P_{pub}, H_0, H_1)$ . The master key is  $s$ .  $H_0$  and  $H_1$  behave as random oracles [1].

**Extract.** Given an identity  $ID$ , the algorithm computes  $D_{ID} = sH_0(ID)$  and output it as the private key related to  $ID$  corresponding to  $Q_{ID} = H_0(ID)$ .

**OffSign.** Given a secret key  $D_{ID}$ , pick a random number  $r \in \mathbb{Z}_q^*$  and a random secret number  $x \in \mathbb{Z}_q^*$ , output the offline signature pair  $(S, R)$ , where  $S = \frac{1}{r}D_{ID}$ ,  $R = xP$ .

**OnSign.** Given a message  $m$  and offline signature  $S$ , compute the online signature as  $\sigma = H_1(m, R)x + r$ . The resulting signature is a triple  $(s, \sigma, R)$ .

**Verify.** Given a signature tuple  $(S, \sigma, R)$  of a message  $m$  for an identity  $ID$ , check whether  $(P_{pub}, \sigma P - H_1(m, R)R, S, Q_{ID})$  is a valid Diffie-Hellman tuple.

### 5.1 Analysis

In this section, we will discuss the correctness and efficiency of our scheme.

**Correctness:** The correctness can be easily proved as follow:

$$\begin{aligned}
 e(\sigma P - H_1(m, R)R, S) &= e(xH_1(m, R)P + rP - H_1(m, R)R, \frac{1}{r}D_{ID}) \\
 &= e(H_1(m, R)xP + rP - H_1(m, R)xP, \frac{1}{r}D_{ID}) \\
 &= e(rP, \frac{1}{r}sQ_{ID}) \\
 &= e(P_{pub}, Q_{ID})
 \end{aligned}$$

**Signature Size:** The resulting signature is the tripe  $(S, \sigma, R)$ . We assume the safe length of GDH group  $\mathbb{G}_1$  is  $\rho$ , the size of each element in a signature tuple is  $\log_2 \rho$ ,  $\log_2 q$ , and  $\log_2 \rho$ . Therefore, the total length is  $2 \log_2 \rho + \log_2 q$ . We believe this size is irreducible since the first two elements are required in all the standard ID-based signature scheme.

**Performance:** Obviously, the online phase of our scheme is very efficient, which only requires one hash, one multiplication, and one addition. The computational workload is passed to the offline phase. The signature verification is done through two pairing operations, which is the most expensive part in our scheme. However, since  $e(P_{pub}, Q_{ID})$  is a constant, it only needs to be computed once.

### 5.2 Security Proof

To prove our scheme is secure against *adaptive chosen message and ID attack*, the problem is firstly reduced to a *given ID attack*. Specifically, we intend to view the scheme as an ordinary ID-based signature scheme which outputs two signatures. Since the online signing phase does not using ID information, it can be viewed as an sub-phase of ID-based offline signing phase.

Specifically, we intend to view the scheme as an ordinary ID-based signature scheme which output two signatures. Since the online signing phase does not using ID information, it is viewed as an sub-phase of ID-based offline signing phase.

**Lemma 1.** *Let  $\mathcal{A}_0$  be an algorithm for an adaptive chosen message and ID attack to our scheme with running time  $t_0$  and advantage  $\epsilon_0$ , then there is an algorithm  $\mathcal{A}_1$  for an adaptive chosen message and given ID attack which has running time  $t_1 \leq t_0$  and advantage  $\epsilon_1 \leq \epsilon_0(1 - \frac{1}{q})/q_{H_0}$ , where  $q_{H_0}$  is the maximum number of queries to ID hash oracle  $H_2$  asked by  $\mathcal{A}_0$ .*

*Proof.* We assume that the number of queries to message hash oracle, extraction oracle and online signing oracle are  $q_{H_1}$ ,  $q_E$ , and  $q_S$ . Algorithm  $\mathcal{A}_1$  is performed as follow:

1. Randomly choose  $l \in \{1, \dots, q_{H_0}\}$ . Let  $ID_i$  denote the input of  $i^{th}$   $q_{H_0}$  query asked by  $\mathcal{A}_0$ . Set  $ID'_i$  be  $ID^*$  if  $i = l$ , and  $ID_i$  otherwise. Define  $H'_0(ID_i)$ ,  $\text{Extract}'(ID_i)$ ,  $\text{Sign}'(ID_i, m)$  to be  $H_0(ID'_i)$ ,  $\text{Extract}(ID'_i)$ ,  $\text{Sign}(ID'_i, m)$ . Notice that the  $\text{Sign}$  includes  $\text{OffSign}$  and  $\text{OnSign}$ . However, only the offline signing part is considered in an ID attack, since the online signing part does not use any ID information.

2. Run  $\mathcal{A}_0$  with the given system parameters.  $\mathcal{A}_1$  responds to  $\mathcal{A}_0$ 's queries to  $H_0, H_1, \text{Extract}$ , and  $\text{Sign}$  by evaluating  $H'_0, H_1, \text{Extract}'$ , and  $\text{Sign}'$ , respectively. Let the output of  $\mathcal{A}_0$  be  $(ID_{out}, m, S, \sigma)$ .
3. If  $ID_{out} = ID^*$  and  $(ID_{out}, m, S, \sigma)$  is valid, the output  $(ID^*, m, S, \sigma)$ . Otherwise output fail.

Since the probability distributions provided by  $H'_0, \text{Extract}'$ , and  $\text{Sign}'$  are indistinguishable from those produced by  $H_0, \text{Extract}$ , and  $\text{Sign}$ ,  $\mathcal{A}_0$  learns nothing from query result. Besides, the probability that  $\mathcal{A}_0$  produces a valid message signature pair  $(ID, m, S, \sigma)$  without any query of  $H'_0(ID)$  is greater than  $(1 - \frac{1}{q})$ . Hence, we can say  $\mathcal{A}_0$  wins the game with advantage  $\geq \epsilon(1 - \frac{1}{q})/q_{H_0}$ , where  $\epsilon$  is an upper bound of success.  $\square$

**Lemma 2.** *If there is an algorithm  $\mathcal{A}_1$  for an adaptive chosen message and given ID attack to our scheme which queries  $H_1, H_2, \text{Sign}$  and  $\text{Extract}$  at most  $q_{H_1}, q_{H_2}, q_S$  and  $q_E$  times respectively, and has running time  $t_1$  and advantage  $\epsilon_1 \geq 10(q_S + 1)(q_S + q_{H_1})/q$ , then CDHP can be solved with probability  $\epsilon_2 \geq 1/9$  within running time  $t_2 \leq 23q_{H_1}t_1/\epsilon_1$ .*

*Proof.* We assume that for any  $ID$ ,  $\mathcal{A}_1$  queries  $H_0(ID)$  and  $\text{Extract}$  at most once. We have an algorithm  $\mathcal{A}_1$ , through interacting with a signing simulator  $\mathcal{B}$ , computes  $abP$  for a randomly given instance  $(P, aP, bP)$  where  $P$  is a generator of  $G$ .

1. Fix an identity  $ID$  and put  $P_{pub} = aP$ . Randomly choose  $\alpha_i \in \mathbb{Z}_q^*$  for  $i = 1, \dots, q_E$  and  $\beta_j, x_j \in \mathbb{Z}_q^*$  for  $j = 1, \dots, q_S$ . Let  $ID_i$  and  $ID_{i_k}$  denote the input of the  $i^{th}$   $H_0$  query and the  $k^{th}$   $\text{Extract}$  query. We define:

$$H''_0(ID) = \begin{cases} bP & \text{if } ID_i = ID^*, \\ \alpha_j P & \text{otherwise;} \end{cases}$$

$$\text{Extract}''(ID_{i_k}) = \alpha_{i_k}(bP);$$

$$\text{OffSign}''(m_j, x_j) = (m_j, h_j, \sigma_j), \text{ where } h_j = H_1(m, R_j), \sigma_j = h_j x_j + \frac{a}{\beta};$$

$$\text{OnSign}''(ID_{i_j}) = (ID_{i_j}, R_j, S_j), \text{ where } R_j = x_j P.$$

The resulting signature is  $(ID_j, m_j, R_j, S_j, \sigma_j)$ . We observed that  $(bP, \sigma P - Rh, S, aP)$  is valid Diffie-Hellman tuple since:

$$\begin{aligned} e((hx + \frac{a}{\beta})P - hR, S) &= e(hxP - \frac{a}{\beta}P - hxP, \beta bP) \\ &= e(\frac{a}{\beta}P, \beta bP) \\ &= e(aP, bP) \end{aligned}$$

2. We apply the oracle replay attack invented by Pointcheval and Stern in [9].
  - (a)  $\mathcal{A}_1$  firstly asks  $q_{H_1}$  distinct queries to the random oracle  $f$ , obtaining  $\rho_1, \dots, \rho_{q_{H_1}}$  answers respectively. Assume there is a simulator  $\mathcal{B}$  which simulates the activity of signer without the knowledge of secret key. For

each query of message  $m_j$  it output a series of signature message pairs in the form of  $(ID_j, m_j, R_j, h_j, S_j, \sigma_j)$ . Then algorithm  $\mathcal{A}_1$  assumes that  $f(m_j, R_j) = h_j$  and stores it.

- (b) If following collisions appear:
    - A  $(m_j, R_j)$  pair produced by  $\mathcal{B}$  also appears in the list of questions to random oracle asked by  $\mathcal{A}_1$ ;
    - $\mathcal{B}$  produces two  $(m_j, R_j)$  pairs which are exactly the same; $\mathcal{A}_1$  simply outputs fail and aborts. If no collision appeared,  $\mathcal{A}_1$  outputs a valid message signature pair, which is expected to be valid for the fixed ID.
  - (c) By replaying  $\mathcal{B}$  with the same messages but different choice of  $H_1$ , we can obtain two valid signatures  $(ID, m, R, h, S, \sigma)$  and  $(ID, m, R, h', S, \sigma')$ , where  $h \neq h'$ . Notice that offline signatures are supposed to be the same since it is closely related to the value of  $r$ .
  - (d) If both outputs are valid, compute  $x = \frac{\sigma - \sigma'}{h - h'}$ .
3. Since  $(Q_{ID_j}, \sigma_j P - R_j h_j, S_j, P_{pub})$  is valid Diffie-Hellman tuple, we can compute  $\alpha$  through  $\beta = \frac{a}{\sigma - h_j x_j}$ . Apply  $\beta_j$  to  $S_j$ , we have

$$\begin{aligned}
 S &= \frac{a}{\sigma - h_j x_j} (bP) \\
 S &= \frac{abP}{\sigma - h_j x_j} \\
 abP &= S(\sigma - h_j x_j) \quad \square
 \end{aligned}$$

Combining Lemma 1 and 2, we have the following theorems.

**Theorem 1.** *If there is an algorithm  $\mathcal{A}_0$  for an adaptive chosen message and ID attack to our scheme which queries  $H_0, H_1, \text{Sign}$  and  $\text{Extract}$  at most  $q_{H_0}, q_{H_1}, q_S$  and  $q_E$  times respectively, and has running time  $t_1$  and advantage  $\epsilon_0 \geq 10(q_S + 1)(q_S + q_{H_1})q_{H_0}/(q - 1)$ , then CDHP can be solved with probability  $\geq 1/9$  within running time  $\leq \frac{23q_{H_0}q_{H_1}t_0}{\epsilon_0(1 - \frac{1}{q})}$ .*

Using another variant of the forking lemma [9], we have the following result:

**Theorem 2.** *If there is an algorithm  $\mathcal{A}_1$  for an adaptive chosen message and given ID attack to our scheme which queries  $H_0, H_1, \text{Sign}$  and  $\text{Extract}$  at most  $q_{H_0}, q_{H_1}, q_S$  and  $q_E$  times respectively, and has running time  $t_1$  and advantage  $\epsilon_1 \geq 10(q_S + 1)(q_S + q_{H_1})/q$ , then CDHP can be solved within expected time  $\leq 120686q_{H_1}t_1/\epsilon_1$ .*

**Theorem 3.** *If there is an algorithm  $\mathcal{A}_0$  for an adaptive chosen message and ID attack to our scheme which queries  $H_0, H_1, \text{Sign}$  and  $\text{Extract}$  at most  $q_{H_0}, q_{H_1}, q_S$  and  $q_E$  times respectively, and has running time  $t_1$  and advantage  $\epsilon_1 \geq 10(q_S + 1)(q_S + q_{H_1})q_{H_0}/(q - 1)$ , then CDHP can be solved within expected time  $\leq \frac{120686q_{H_0}q_{H_1}t_0}{\epsilon_0(1 - \frac{1}{q})}$ .*

## 6 Application for AODV

We now describe how to apply our scheme to the AODV routing protocol, which is quite straightforward and its security can be significantly reinforced by merely using digital signature. We will briefly introduce the security requirement of AODV routing protocol and then describe how our scheme can be implemented over it.

### 6.1 AODV Security Requirement

AODV is a simple and efficient on-demand ad hoc routing protocol. Basically, it uses RREQ (route request), RREP (route reply) and RRER (route error) messages to accomplish route discovery and maintenance operations. It also utilizes sequence numbers to prevent routing loops. Routing decision making is based on sequence numbers and routes maintained in each node's routing table.

We require that each node must submit its identity to the key generation center before entering the network through a secure channel. The key generation center will generate a private key correspondent to node's ID, and send it to the node along with necessary system parameters. In an ad hoc environment, this phase should be performed offline.

After entering the network, each node starts to compute its offline signature. Since the offline signature is created over a random value, the node can randomly choose several values and compute the signatures respectively for each session. When a routing request is initiated, the node generates a routing packet (RREQ/RREP) according to AODV and generate the online signature for this packet. This phase is very efficient since signature generation only requires one hash. Then the sender node broadcasts the packet and signature to neighbors.

When a neighboring node receives this packet, it will verify this signature and broadcast to the next hop. To be efficient, the verification can be done offline. The receiving node should broadcasts the packet before verification. However, only if this packet passes the verification, will the receiving update its routing table entry according to the information carried in the packet.

By deploying our scheme, the efficiency of SAODV can be improved. This scheme can also be used in some other routing protocol such as DSR, which requires much more frequent signing operations. Using bilinear pairing would engender the major cost in our scheme, but the realization of ID-based authentication scheme can largely solve the diehard key distribution problem in MANET.

## 7 Conclusion

We proposed an ID-based online/offline signature scheme from bilinear pairings that is suitable for MANET. In our scheme, the resulting signature is a triplet. The online signature can be computed is very efficient, approximately one hash operation. The computation of the offline phase requires only one scalar multiplication under an additive group. The verification is done through pairings

but its performance can be enhanced after the first execution. We proved that our scheme is secure against existential forgery under adaptive chosen message attacks based on the random oracle model. The security of our scheme is based on CDHP. Our scheme is especially suitable for mobile ad hoc networks routing where signature enabled authentication is to be performed in an efficient manner. We also discussed the implementation issue over MANET routing protocols and presented an implementation method over AODV routing protocol.

## References

1. M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the First Annual Conference on Computer and Communications Security*. ACM, November 1993.
2. J. Cha and J. Cheon. An id-based signature from gap-diffie-hellman groups. In *Proceedings of Public Key Cryptography - PKC 2003*, volume 2567, pages 1–24. Springer-Verlag, 2003.
3. D. Boneh, B. Lynn, and H. Shacham. Short signature from the weil pairing. In *Proceedings of Asiacrypt '01, Lecture Notes in Computer Sciences*, volume 2248, pages 514–532. MANET working group, 2001.
4. S. Even, O. Goldreich, and S. Micali. On-line/off-line digital signatures. In *Proceedings of Advances in Cryptology: Crypto '89*. Springer, 1990.
5. S. Glodwasser, S. Micali, and R. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal of Computing*, 17:281–308, 1988.
6. Y. C. Hu, A. Perrig, and D. B. Johnson. Ariadne: A secure on-demand routing protocol for ad hoc networks. In *Proceedings of the Eighth ACM International Conference on Mobile Computing and Networking (Mobicom 2002)*, 2002.
7. D. B. Johnson, D. A. Maltz, and Y. C. Hu. *The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)*, 2004.
8. C. E. Perkins, E. M. Royer, and S. R. Das. *Ad Hoc On-Demand Distance Vector (AODV) Routing*, 2003.
9. D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13:361–396, 2000.
10. M. G. Zapata. *Secure Ad hoc On-Demand Distance Vector (SAODV) Routing*, 2004.
11. F. Zhang, Y. Mu, and W. Susilo. Reducing security overhead for mobile networks. In *Proceedings of The 19th International Conference on Advanced Information Networking and Applications (AINA 2005)*, pages 398–403. IEEE Computer Society, 2005.

# Collision Attack on XTR and a Countermeasure with a Fixed Pattern\*

Dong-Guk Han<sup>1</sup>, Tsuyoshi Takagi<sup>1</sup>, Tae Hyun Kim<sup>2</sup>,  
Ho Won Kim<sup>3</sup>, and Kyo Il Chung<sup>3</sup>

<sup>1</sup> Future University-Hakodate, Japan  
{christa, takagi}@fun.ac.jp

<sup>2</sup> Center for Information and Security Technologies(CIST),  
Korea University, Seoul, Korea

thkim@cist.korea.ac.kr

<sup>3</sup> Electronics and Telecommunications Research Institute(ETRI), Korea  
{khw, kyoil}@etri.re.kr

**Abstract.** Recently, XTR is considered as one of good candidates for more energy efficient cryptosystems. Among the family of XTR algorithms, the Improved XTR Single Exponentiation (XTR-ISE) is the most efficient one suitable for ubiquitous computer. Even though the security of such devices against side channel attacks is very dangerous, there are few works on side channel attacks against XTR-ISE. In this paper we propose a new collision attack on XTR-ISE. The analysis complexity of the proposed one is about  $2^{40}$  where the key size is 160-bit, which is 55% improvement from the previously best known analysis of Page-Stam. We also propose a novel countermeasure using a fixed pattern which is secure against SPA. In the sense of both efficiency and security the proposed countermeasure is the best one among the previous countermeasures- it is about 30% faster.

**Keywords:** Ubiquitous computer, XTR public key system, XTR Exponentiation Algorithms, Side Channel Attacks, Collision Attack.

## 1 Introduction

We are standing to the beginning of the ubiquitous computing era. It is expected that we can accomplish lucrative applications by effectively synthesizing the ubiquitous computer with cryptography. The ubiquitous computer only has scarce computational resources (like Smart cards, RFID, Sensor Network), so that we have to make an effort to optimize the memory and efficiency of the security system. Currently there are a few implementations on ubiquitous environments with PKC. In ESAS 2004 Gaubatz-Kaps-Sunar showed an implementation of Rabin and Ntru in sensor networks [6]. Recently Watro et al. showed RSA (in the case the encryption key is 3) is feasible to the applications of ubiquitous computer, and remarked that XTR is one of good candidates for light

---

\* The full version of this paper was posted in the Cryptology ePrint Archive [9].



weight cryptosystems in SASN 2004 [18]. However, the applications of ubiquitous computer will be carried into and used in hostile environments and often house sensitive information, for example identity related tokens or financial information, the threat of attack is significant. This threat is magnified by both the potential pay-off and level of anonymity that side channel attacks (SCA) allow [10, 11]. The fact that one can attack a device somewhat remotely via timing and power consumption means that most ubiquitous computing devices need to be aware of similar problems in their operational environments.

In Crypto 2000 Lenstra-Verheul introduced XTR public key cryptosystems [12]. Given the current state of affairs in breaking the discrete logarithm problems over either finite fields or elliptic curves, XTR can compete with elliptic curve cryptosystems (ECC) in terms of both speed and bandwidth. This makes XTR suitable for deployment on similar sorts of constrained devices such as smart-cards, where computational power and storage capacity are both very limited. Among the family of XTR exponentiation algorithms, the Improved XTR Single Exponentiation (XTR-ISE) is the most efficient one suitable for smart-cards, where computational power and memory capacity are both very limited. Even though the security of such devices against side channel attacks is very dangerous, however, there are few works on side channel attacks against XTR-ISE.

In 2004 Chung-Hasan [2] and Page-Stam [14] proposed simple power analysis (SPA) against XTR-ISE and that it was the first try to analyze it with SCA. Chung-Hasan showed it takes  $2^{100}$  tries for an attacker until he/she correctly finds the secret key in XTR-ISE with 160-bits key length. On the other hand, Page-Stam showed it requires  $2^{88}$  tries. However, these results are far worse than well-known square-root type algorithms (Baby-Step-Giant-Step or Pollards' Rho methods).

In this paper we find a new analysis technique, called as XTR collision attack, derived from the structural properties of XTR-ISE. The complexity of XTR collision attack is about  $2^{0.25 \cdot l}$  where  $l$  is the length of the key, which is about 55% improvement from the result of Page-Stam [14]. Also we propose a novel countermeasure using a fixed pattern which is secure against SPA. In the sense of both efficiency and security the proposed countermeasure is the best one among the previous countermeasures- it is about 30% faster.

## 2 XTR Exponentiation Algorithm

In this section, we review the fundamental algorithms to calculate traces of powers [12, 15]. For an element  $h \in \mathbf{F}_{p^2}^*$  its trace  $\text{Tr}(h)$  over  $\mathbf{F}_{p^2}$  is defined as a sum of the conjugates over  $\mathbf{F}_{p^2}$  of  $h$ :  $\text{Tr}(h) = h + h^{p^2} + h^{p^4} \in \mathbf{F}_{p^2}$ . Throughout this paper,  $c_n$  denotes  $\text{Tr}(g^n) \in \mathbf{F}_{p^2}$ , for some fixed  $p$  and  $g$  of order  $q$ , where  $q$  divides  $p^2 - p + 1$ . Note that  $c_0 = 3$  and  $c_1 = c$ .

An efficient computation of  $c_n$  for given  $p, q$  and  $c$  depends on the recurrence relations  $c_{u+v} = c_u c_v - c_v^p c_{u-v} + c_{u-2v}$ , and  $c_{2u} = c_u^2 - 2c_u^p$ , which is derived from the previous one when  $u = v$ .

By using above two formulae, we define the following two functions called as XTR addition and XTR doubling respectively;

$$A[x, y, z, w] = x \cdot y - y^p \cdot z + w,$$

$$D[x] = x^2 - 2x^p.$$

By using above defined notations we introduce Improved XTR exponentiation algorithms proposed by Stam-Lenstra [15]. The goal of these algorithms is to compute  $c_n$  for given  $c_1$  and  $n \in \mathbb{Z}$ , i.e. to compute  $Tr(g^n)$  with  $Tr(g)$  and an integer  $n$ .

**Improved XTR Single Exponentiation (XTR-ISE) [15]**

Input:  $c_1$  and  $n$  where  $n > 2$

Output:  $c_n$

1. Initialization:
  - 1.1. Let  $a = \text{round}(\frac{3-\sqrt{5}}{2}n)$  and  $b = n - a$  (where  $\text{round}(x)$  is the integer closest to  $x$ ).
  - 1.2. Let  $f = 0$ . As long as  $a$  and  $b$  are both even, replace  $(a, b)$  by  $(a/2, b/2)$  and  $f$  by  $f + 1$ .
  - 1.3. Let  $i = 1$  and  $G_i := (Q_0, Q_1, Q_2, Q_3) = (c_1, c_1, 3, c_1^p)$ .
2. As long as  $a \neq b$ 
  - 2.1. If  $b > a$ 

<ol style="list-style-type: none"> <li>X<sub>1</sub>. if <math>b \leq 4a</math>, then <math>(a, b) \leftarrow (b - a, a)</math> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;"><math>T_0 \leftarrow A[Q_0, Q_1, Q_2, Q_3],</math></td> <td style="width: 50%;"><math>T_1 \leftarrow Q_0,</math></td> </tr> <tr> <td><math>T_2 \leftarrow Q_1,</math></td> <td><math>T_3 \leftarrow Q_2^p.</math></td> </tr> </table> </li> <li>X<sub>2</sub>. else if <math>b</math> is even, then <math>(a, b) \leftarrow (a, b/2)</math> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;"><math>T_0 \leftarrow D[Q_0],</math></td> <td style="width: 50%;"><math>T_1 \leftarrow Q_1,</math></td> </tr> <tr> <td><math>T_2 \leftarrow A[Q_0, Q_2, Q_1, Q_3^p],</math></td> <td><math>T_3 \leftarrow D[Q_2].</math></td> </tr> </table> </li> <li>X<sub>3</sub>. else if <math>a</math> is odd, then <math>(a, b) \leftarrow (a, (b - a)/2)</math> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;"><math>T_0 \leftarrow D[Q_0],</math></td> <td style="width: 50%;"><math>T_1 \leftarrow A[Q_0, Q_1, Q_2, Q_3],</math></td> </tr> <tr> <td><math>T_2 \leftarrow Q_2,</math></td> <td><math>T_3 \leftarrow D[Q_1]^p.</math></td> </tr> </table> </li> <li>X<sub>4</sub>. else (<math>a</math> is even), then <math>(a, b) \leftarrow (b, a/2)</math> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;"><math>T_0 \leftarrow D[Q_1],</math></td> <td style="width: 50%;"><math>T_1 \leftarrow Q_0,</math></td> </tr> <tr> <td><math>T_2 \leftarrow Q_3^p,</math></td> <td><math>T_3 \leftarrow D[Q_2]^p.</math></td> </tr> </table> </li> </ol>	$T_0 \leftarrow A[Q_0, Q_1, Q_2, Q_3],$	$T_1 \leftarrow Q_0,$	$T_2 \leftarrow Q_1,$	$T_3 \leftarrow Q_2^p.$	$T_0 \leftarrow D[Q_0],$	$T_1 \leftarrow Q_1,$	$T_2 \leftarrow A[Q_0, Q_2, Q_1, Q_3^p],$	$T_3 \leftarrow D[Q_2].$	$T_0 \leftarrow D[Q_0],$	$T_1 \leftarrow A[Q_0, Q_1, Q_2, Q_3],$	$T_2 \leftarrow Q_2,$	$T_3 \leftarrow D[Q_1]^p.$	$T_0 \leftarrow D[Q_1],$	$T_1 \leftarrow Q_0,$	$T_2 \leftarrow Q_3^p,$	$T_3 \leftarrow D[Q_2]^p.$
$T_0 \leftarrow A[Q_0, Q_1, Q_2, Q_3],$	$T_1 \leftarrow Q_0,$															
$T_2 \leftarrow Q_1,$	$T_3 \leftarrow Q_2^p.$															
$T_0 \leftarrow D[Q_0],$	$T_1 \leftarrow Q_1,$															
$T_2 \leftarrow A[Q_0, Q_2, Q_1, Q_3^p],$	$T_3 \leftarrow D[Q_2].$															
$T_0 \leftarrow D[Q_0],$	$T_1 \leftarrow A[Q_0, Q_1, Q_2, Q_3],$															
$T_2 \leftarrow Q_2,$	$T_3 \leftarrow D[Q_1]^p.$															
$T_0 \leftarrow D[Q_1],$	$T_1 \leftarrow Q_0,$															
$T_2 \leftarrow Q_3^p,$	$T_3 \leftarrow D[Q_2]^p.$															
  - 2.2. Else (if  $a > b$ )
 

<ol style="list-style-type: none"> <li>Y<sub>1</sub>. if <math>a \leq 4b</math>, then <math>(a, b) \leftarrow (a - b, b)</math> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;"><math>T_0 \leftarrow A[Q_0, Q_1, Q_2, Q_3],</math></td> <td style="width: 50%;"><math>T_1 \leftarrow Q_1,</math></td> </tr> <tr> <td><math>T_2 \leftarrow Q_0,</math></td> <td><math>T_3 \leftarrow Q_2.</math></td> </tr> </table> </li> <li>Y<sub>2</sub>. else if <math>a</math> is even, then <math>(a, b) \leftarrow (b, a/2)</math> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;"><math>T_0 \leftarrow D[Q_1],</math></td> <td style="width: 50%;"><math>T_1 \leftarrow Q_0,</math></td> </tr> <tr> <td><math>T_2 \leftarrow Q_3^p,</math></td> <td><math>T_3 \leftarrow D[Q_2]^p.</math></td> </tr> </table> </li> <li>Y<sub>3</sub>. else if <math>b</math> is odd, then <math>(a, b) \leftarrow (b, (a - b)/2)</math> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;"><math>T_0 \leftarrow D[Q_1],</math></td> <td style="width: 50%;"><math>T_1 \leftarrow A[Q_0, Q_1, Q_2, Q_3],</math></td> </tr> <tr> <td><math>T_2 \leftarrow Q_2^p,</math></td> <td><math>T_3 \leftarrow D[Q_0]^p.</math></td> </tr> </table> </li> <li>Y<sub>4</sub>. else (<math>b</math> is even), then <math>(a, b) \leftarrow (a, b/2)</math> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;"><math>T_0 \leftarrow D[Q_0],</math></td> <td style="width: 50%;"><math>T_1 \leftarrow Q_1,</math></td> </tr> <tr> <td><math>T_2 \leftarrow A[Q_0, Q_2, Q_1, Q_3^p],</math></td> <td><math>T_3 \leftarrow D[Q_2]</math></td> </tr> </table> </li> </ol>	$T_0 \leftarrow A[Q_0, Q_1, Q_2, Q_3],$	$T_1 \leftarrow Q_1,$	$T_2 \leftarrow Q_0,$	$T_3 \leftarrow Q_2.$	$T_0 \leftarrow D[Q_1],$	$T_1 \leftarrow Q_0,$	$T_2 \leftarrow Q_3^p,$	$T_3 \leftarrow D[Q_2]^p.$	$T_0 \leftarrow D[Q_1],$	$T_1 \leftarrow A[Q_0, Q_1, Q_2, Q_3],$	$T_2 \leftarrow Q_2^p,$	$T_3 \leftarrow D[Q_0]^p.$	$T_0 \leftarrow D[Q_0],$	$T_1 \leftarrow Q_1,$	$T_2 \leftarrow A[Q_0, Q_2, Q_1, Q_3^p],$	$T_3 \leftarrow D[Q_2]$
$T_0 \leftarrow A[Q_0, Q_1, Q_2, Q_3],$	$T_1 \leftarrow Q_1,$															
$T_2 \leftarrow Q_0,$	$T_3 \leftarrow Q_2.$															
$T_0 \leftarrow D[Q_1],$	$T_1 \leftarrow Q_0,$															
$T_2 \leftarrow Q_3^p,$	$T_3 \leftarrow D[Q_2]^p.$															
$T_0 \leftarrow D[Q_1],$	$T_1 \leftarrow A[Q_0, Q_1, Q_2, Q_3],$															
$T_2 \leftarrow Q_2^p,$	$T_3 \leftarrow D[Q_0]^p.$															
$T_0 \leftarrow D[Q_0],$	$T_1 \leftarrow Q_1,$															
$T_2 \leftarrow A[Q_0, Q_2, Q_1, Q_3^p],$	$T_3 \leftarrow D[Q_2]$															
  - 2.3.  $i \leftarrow i + 1$  and set  $G_i = (T_0, T_1, T_2, T_3)$ .
3. Compute  $\tilde{c} = A[Q_0, Q_1, Q_2, Q_3] = c_{u+v}$ .
4. Output  $\tilde{c}_{2^f}$ .
5. If  $a = 1$  then return  $\tilde{c}_{2^f}$   
 else run Improved XTR Single Exponentiation with  $c = \tilde{c}_{2^f}$  and  $n = a$ .

### 3 New Collision Attack on XTR

In this section we find a new analysis technique, called as XTR collision attack, derived from the structural properties of XTR-ISE.

### 3.1 Some Properties of XTR-ISE

In XTR-ISE, Step 2 consists of eight states  $X_i$  and  $Y_i$  where  $1 \leq i \leq 4$ . One state is only determined by the condition of  $a$  and  $b$ . From XTR-ISE we can derive the following finite markov chain depicted in figure 1.

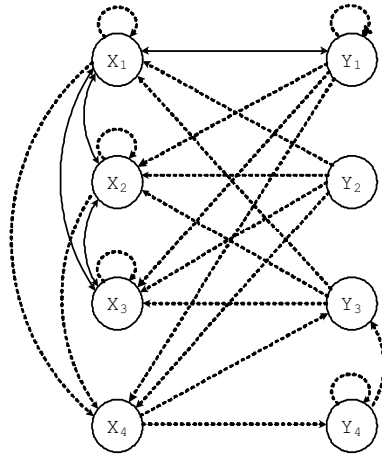


Fig. 1. The finite Markov chain associated with XTR-ISE

*Property 1.* State  $Y_2$  never occurs in the process of XTR-ISE except the first time.

### 3.2 Assumptions and Notations

We first introduce some reasonable assumptions which are used in a new attack.

1.  $A[x, y, z, w]$  and  $D[x]$  are distinguishable by a single measurement of power consumption, whereas  $D[x]^p$  and  $D[x]$ , and  $A[x, y, z, w^p]$  and  $A[x, y, z, w]$  are indistinguishable, respectively. Here,  $x, y, z, w \in \mathbf{F}_{p^2}$ .
2. When  $\{A[\cdot], D[\cdot], D[\cdot]\}$  are all operated, e.g. in the case of  $X_3$  in XTR-ISE, we assume these three functions are operated according to the following order  $A[\cdot]D[\cdot]D[\cdot]$ . In more detail, states  $X_i$  and  $Y_i$  are updated according to the following orders;
  - (a) In  $X_2$  and  $Y_4$ : the computation order is  $T_2 \rightarrow T_0 \rightarrow T_3 \rightarrow T_1$ ,
  - (b) In  $X_3$  and  $Y_3$ : the computation order is  $T_1 \rightarrow T_0 \rightarrow T_3 \rightarrow T_2$ .
3. If  $D[c_u]$  and  $D[c_v]$  are computed, the attacker is not able to guess the value of  $c_u$  nor  $c_v$  but he/she is able to check if  $c_u = c_v$ .

As the required computing time of  $A[\cdot]$  is two times of that of  $D[\cdot]$  and  $p$ -th powering is free (refer to [12]) in XTR, the above Assumption 1 is reasonable. Assumption 3 is also reasonable since this kind of computation usually takes

many clock cycles and depends greatly on the value of the operand. This kind assumption has been used in a stronger variant and validated by Schramm et al. [16] who are able to distinguish collisions during one DES round computation. It was extended to ECC by Fouque et al [5].

**Notations:** For simplicity,  $A[x, y, z, w]$  and  $D[x]$  are referred to as **A** and **D**, respectively. Let  $S[c_1, n]$  be an AD sequence when the inputs are  $c_1$  and  $n$  in XTR-ISE, i.e. **A** and **D** are elements of  $S[c_1, n]$ , which are written with time-increasing from left to right. Due to the above Assumption 1, **A** and **D** also denote  $A[x, y, z, w^p]$  and  $D[x]^p$ , respectively.

As described in XTR-ISE algorithm,  $G_i = (Q_0, Q_1, Q_2, Q_3)$  is the  $i$ -th updated intermediate values of  $\{Q_j\}_{0 \leq j \leq 3}$  in Step 2 of XTR-ISE for  $i \geq 1$ . For  $G_i \xrightarrow{T_i} G_{i+1}$  where  $T_i \in \{X_j, Y_j\}_{1 \leq j \leq 4}$ , if  $T_i$  is one of  $\{X_2, X_3, X_4, Y_3, Y_4\}$ , then **DD** is computed. We denote these two **DD** as  $\mathbf{D}_1^1 \mathbf{D}_1^2$  ( $\mathbf{D}_1^1 \mathbf{D}_1^2$  are carried along for expository purposely only).

### 3.3 Attacker’s Goal

In XTR-ISE, Step 2 consists of eight states  $X_i$  and  $Y_i$  where  $1 \leq i \leq 4$ . One state is only determined by the condition of two integers  $a$  and  $b$ . However, if an attacker can decide the states executed during the computation then the secret key can be easily reconstructed from the recovered state.

Under Assumption 1, an attacker is able to distinguish **A**, **DD**, and **ADD**. With this information he/she can categorize seven states of XTR-ISE into the following three groups;

- **A** is corresponding to  $X_1$  or  $Y_1$ ,
- **DD** is corresponding to  $X_4$ ,
- **ADD** is corresponding to  $X_2, X_3, Y_3$  or  $Y_4$ .

However, there are some ambiguity decisions such as (1)  $X_1$  and  $Y_1$  are not distinguished, (2) if **ADD** is observed in AD sequence then there are two possibilities; **ADD** and **A|DD**. Using a brute force search technique, one might test around 6 candidates; i.e. **ADD** is corresponding to one of  $\{X_2, X_3, Y_3, Y_4, X_1|X_4, Y_1|X_4\}$ .

Thus the attacker needs to check the possible candidates until he/she has found the correct one, so in order to improve the efficiency of the attack we want to minimize the number of candidates.

### 3.4 Analysis Based on the Finite Markov Chain

First we consider the following three types of AD sequences;

- $\mathbf{ADD|DD}$ .
- $\overbrace{\mathbf{ADD|ADD} \dots \mathbf{ADD}}^{m\text{-times}}$ , briefly it is denoted as  $\{\mathbf{ADD}\}^m$ .
- $\mathbf{ADD|}\overbrace{\mathbf{A} \dots \mathbf{A}}^{m\text{-times}}|\mathbf{ADD}$ , denoted as  $\mathbf{ADD|}\{\mathbf{A}\}^m|\mathbf{ADD}$ .

When  $\text{ADD}|\text{DD}$  is observed in AD sequence we can decide  $\underbrace{\text{ADD}}_{X_2} | \underbrace{\text{DD}}_{X_4}$ .

Because the last two DD originates from  $X_4$  and the possible preconditions of  $X_4$  are  $X_1, X_2$ , and  $Y_1$ . Thus ADD implies  $X_2$

When  $\{\text{ADD}\}^m$  is observed in AD sequence there are  $6^m$  possible combinations from  $\{X_2, X_3, Y_3, Y_4, X_1|X_4, Y_1|X_4\}$ . However, if we consider the finite markov chain (Fig. 1) then we can reduce the possible number of combinations such as 15 and 39 combinations when  $m$  is 2 and 3, respectively. If  $m \geq 4$  then the number of all possible combinations from the finite markov chain is  $\#\{[\text{ADD}]^m\} = (39m + 48) \cdot 2^{m-5}$ .

When  $\text{ADD}|\{\text{A}\}^m|\text{ADD}$  is observed in AD sequence there are  $6 \cdot 2^m \cdot 6 = 9 \cdot 2^{m+2}$  combinations of XTR states. However, if we consider the finite markov chain (Fig. 1) then the number of possible combinations is  $3 \cdot 2^{m+1}$ . Furthermore, we propose the following decision rule derived from the finite markov chain (Fig. 1).

*Property 2.* If  $\text{AADDAA}$  is observed in AD sequence then we can decide  $\text{A} | \underbrace{\text{ADD}}_{X_2 \text{ or } X_3} | \underbrace{\text{A}}_{X_1} | \text{A}$ .

### 3.5 XTR Collision Attack

At the previous section, the number of possible combinations for  $\{\text{ADD}\}^m$  and  $\text{ADD}|\{\text{A}\}^m|\text{ADD}$  is decreased by using the finite markov chain of XTR-ISE. In this section, in order to reduce the search space from the finite markov chain we introduce a new attack mainly based on the above assumptions, especially Assumption 3, described in 3.2.

**Key Observation:** If we focus on D operation, we notice that some of them manipulate the same operand. We consider two AD sequences  $S[c_1, n]$  and  $S[c_2, n]$ .

**In the case of  $\{\text{ADD}\}^m$ :** For simplicity, we assume  $m = 2$ , i.e.  $\text{ADDADD}$  is considered. Note that there are 15 combinations of states.

$$S[c_1, n] = \dots \text{AD}_i^1 \text{D}_i^2 \text{AD}_j^1 \text{D}_j^2 \dots$$

$$S[c_2, n] = \dots \text{AD}_i^1 \text{D}_i^2 \text{AD}_j^1 \text{D}_j^2 \dots$$

Depending on the combination type, we can observe the following results;

**CASE\_I:**  $\text{D}_j^1$  of  $S[c_1, n]$  is same to  $\text{D}_i^1$  of  $S[c_2, n]$ ,

**CASE\_II:**  $\text{D}_j^2$  of  $S[c_1, n]$  is same to  $\text{D}_i^1$  of  $S[c_2, n]$ .

According to the above observation, the 15 combination pairs are categorized as

**CASE\_I:**  $(X_2, X_2), (X_2, X_3), (X_3, X_2), (X_3, X_3), (X_1, X_4, Y_4), (Y_1, X_4, Y_4), (Y_3, X_2), (Y_3, X_3), (Y_4, Y_4), (X_2, X_1, X_4), (X_3, X_1, X_4), (Y_3, X_1, X_4)$ ,

**CASE\_II:**  $(X_1, X_4, Y_3), (Y_1, X_4, Y_3), (Y_4, Y_3)$ .

With this collision information, we can make the following comparison table.

# of all possible combinations			
$m$	From Exhaustive Search	From the Finite Markov Chain	From Collision Attack
1	6	6	6
2	36	15	10.2
3	216	39	17.77
⋮	⋮	⋮	⋮

From the results of the above table we can see that the average number of trial tests with collision information is drastically decreased compared to that of the finite markov chain.

**In the case of  $ADD|A^m|ADD$ :** Consider

$$S[c_1, n] = \dots AD_1^1 D_1^2 \{A\}^m AD_j^1 D_j^2 \dots$$

$$S[c_2, n] = \dots AD_1^1 D_1^2 \{A\}^m AD_j^1 D_j^2 \dots$$

Similar to the previous analysis, we can observe the following results depending on the combination type;

**CASE\_0:** There is no relation between **D** operation of  $S[c_1, n]$  and  $S[c_2, n]$ ,

**CASE\_I:**  $D_j^1$  of  $S[c_1, n]$  is same to  $D_i^1$  of  $S[c_1, n]$ ,

**CASE\_II:**  $D_j^2$  of  $S[c_1, n]$  is same to  $D_i^1$  of  $S[c_2, n]$ .

The results of the following table show the improvement of analysis complexity.

# of all possible combinations			
$m$	From Exhaustive Search	From the Finite Markov Chain	From Collision Attack
1	72	12	4.5
2	144	24	9.75
3	288	48	28.87
⋮	⋮	⋮	⋮

**Implementation Results:** From these classifications, we can reduce the search space order required to detect the whole secret value. From our implementation results the average number of trial XTR exponentiations is roughly given by  $2^{0.25 \cdot l}$  where  $l$  is the length of the exponents. Thus the complexity of XTR collision attack against XTR-ISE is about  $2^{40}$  where the key length is 160-bit, which is about 55% improvement from the result of Page-Stam [14].

### 4 Proposed Countermeasure

In this section we explain the proposed algorithm. We modify XTR-ISE to be secure against SCA. The main idea is same to that of Okeya-Takagi scheme [13] for elliptic curve cryptosystems. In XTR-ISE there are three different patterns, **A**, **DD**, and **ADD**. For example, if  $X_1, Y_1,$  and  $X_4$  are consecutively operated then the sequence is “**AAADD**”, which is no longer the fixed pattern.

We try to generate a XTR operation sequence that has a fixed pattern such that  $|ADD|ADD|\dots|ADD|$ .

---

**Fixed Pattern XTR Single Exponentiation (XTR-FSE)**


---

Input:  $c_1$  and  $n$  where  $n > 2$ Output:  $c_n$ 

- 
1. Initialization:
    - 1.1. Select a random number  $a$  in  $[1, n-1]$  and  $b = n - a$ . If  $a$  is even, then let  $a \leftarrow a + 1$ ,  $b \leftarrow b - 1$ .
    - 1.2. Let  $Q_0 = c$ ,  $Q_1 = c$ ,  $Q_2 = 3$ , and  $Q_3 = c^p$ .
  2. As long as  $a \neq b$ 
    - 2.1. If  $b > a$ 
      - $\mathcal{X}_1$ . if  $b$  is even, then  $(a, b) \leftarrow (a, b/2)$ 

$$\begin{array}{ll} T_0 \leftarrow D[Q_0], & T_1 \leftarrow Q_1, \\ T_2 \leftarrow A[Q_0, Q_2, Q_1, Q_3^p], & T_3 \leftarrow D[Q_2]. \end{array}$$
      - $\mathcal{X}_2$ . else ( $b$  is odd), then  $(a, b) \leftarrow (a, (b-a)/2)$ 

$$\begin{array}{ll} T_0 \leftarrow D[Q_0], & T_1 \leftarrow A[Q_0, Q_1, Q_2, Q_3], \\ T_2 \leftarrow Q_2, & T_3 \leftarrow D[Q_1]^p. \end{array}$$
    - 2.2. Else (if  $a > b$ )
      - $\mathcal{Y}_1$ . if  $b$  is odd, then  $(a, b) \leftarrow (b, (a-b)/2)$ 

$$\begin{array}{ll} T_0 \leftarrow D[Q_1], & T_1 \leftarrow A[Q_0, Q_1, Q_2, Q_3], \\ T_2 \leftarrow Q_2^p, & T_3 \leftarrow D[Q_0]^p. \end{array}$$
      - $\mathcal{Y}_2$ . else ( $b$  is even), then  $(a, b) \leftarrow (a, b/2)$ 

$$\begin{array}{ll} T_0 \leftarrow D[Q_0], & T_1 \leftarrow Q_1, \\ T_2 \leftarrow A[Q_0, Q_2, Q_1, Q_3^p], & T_3 \leftarrow D[Q_2]. \end{array}$$
    - 2.3. Set  $Q_0 \leftarrow T_0$ ,  $Q_1 \leftarrow T_1$ ,  $Q_2 \leftarrow T_2$ ,  $Q_3 \leftarrow T_3$ .
  3. Compute  $\tilde{c} = A[Q_0, Q_1, Q_2, Q_3] = c_{u+v}$ .
  4. If  $a = 1$  then return  $\tilde{c}$ ,  
 else goto Step 1. with  $c = \tilde{c}$  and  $n = a$ .
- 

We can prove the following proposition about the efficiency of XTR-FSE.

**Proposition 1.** *For a given integer  $n$ , the proposed algorithm takes on average  $1.41 \log_2 n$  iterations in Step 2. Thus the trace value  $c_n$  can on average be computed in about  $11.2 \log_2(n)$  multiplications in  $\mathbf{F}_p$  because each step requires 8 multiplications in  $\mathbf{F}_p$  [12].*

#### 4.1 Security Analysis

In this section we discuss the security of the proposed scheme against SPA and DPA.

**SPA:** As we mentioned in the previous section, the proposed method compute XTR single exponentiation through the fixed pattern  $|\mathbf{ADD}|\mathbf{ADD}|\dots|\mathbf{ADD}|$ . The attacker could distinguish XTR operations  $D[\cdot]$  and  $A[\cdot]$  in XTR-FSE by measurement of the power consumption, but he/she obtains only the identical **ADD** sequence for any input  $c$  and  $n$ . Therefore, he/she cannot detect the secret scalar  $n$  by using SPA.

**DPA:** The use of scalar randomization such as exponent splitting [3] prevents against DPA. Note that the idea of splitting the data was already abstracted in [4] as a general countermeasure against DPA. The proposed method is using exponent splitting technique as a DPA countermeasure, i.e. we split the input integer  $n$  into two parts by picking a random  $a \in [1, n-1]$  and rewriting the integer  $n$  as  $a + (n-a)$ . Thus XTR-FSE is secure against DPA.

## 4.2 Comparison of Empirical Performance and Type of Countermeasure

In this section we compare the computational cost and the type of countermeasures between the proposed countermeasure and the previous ones.

The compared three methods use the exponent splitting method as DPA countermeasure. But the utilized SPA countermeasure is different each others. The countermeasure of ICICS'04 is based on XTR-SE. Their method does not require SPA countermeasure because XTR-SE has the fixed operations **ADD**. On the other hand, the countermeasure of SAC'04 and the proposed method is based on XTR-ISE, which does not has fixed operations. In order to solve this problem Page-Stam proposed the indistinguishable arithmetic with dummy operation sometimes, but the security of indistinguishable arithmetic [17] and the dummy method [19] are recently very controversial. From the result of Table 1 our proposed countermeasure is the best one in XTR in the sense of both efficiency and security.

**Table 1.** Comparison of empirical performance and type of countermeasure

	Efficiency	Type of Countermeasure	
	Compute $Tr(g^n)$	SPA	DPA
ICICS'04 [8]	$16 \log_2(n)$	Fixed Pattern + No Dummy Operation	Exponent Splitting
SAC'04 [14]	$8.5 \log_2(n)$	Indistinguishable Assumption + Dummy Operation	Exponent Splitting
Proposed Method	$11.2 \log_2(n)$	Fixed Pattern + No Dummy Operation	Exponent Splitting

## Acknowledgements

Dong-Guk Han was supported by the Korea Research Foundation Grant. (KRF-2005-214-C00016) and Tae Hyun Kim was supported in part by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Assessment).

## References

1. M. Ciet and C. Giraud, *Transient Fault Induction Attacks on XTR*, Information and Communications Security (ICICS 2004), LNCS 3269, (2004), 440-451.
2. J. Chung and A. Hasan, *Security Analysis of XTR Exponentiation Algorithms against Simple Power Analysis Attack*, Preprint of CACR, Univ. of Waterloo, CACR 2004-05.
3. C. Clavier and M. Joye, *Universal Exponentiation Algorithm A First Step towards Provable SPA-Resistance*, Cryptographic Hardware and Embedded Systems (CHES'01), LNCS2162, (2001), 300-308.



4. S. Chari, C.S. Jutla, J.R. Rao, and P. Rohatgi, *Towards sound approaches to counteract power-analysis attacks*, Advances in Cryptology - CRYPTO '99, LNCS1666, (1999), 398-412.
5. P.-A. Fouque and F. Valette, *The Doubling Attack Why Upwards is better than Downwards*, Workshop on Cryptographic Hardware and Embedded Systems 2003 (CHES 2003), LNCS 2779, (2003), 269-280.
6. G. Gaubatz, J.-P. Kaps, and B. Sunar, *Public Key Cryptography in Sensor Networks-Revisited*, 1st European Workshop on Security in Ad-Hoc and Sensor Networks, (ESAS 2004), LNCS3313, (2004), 2-18.
7. R. Granger, D. Page, and M. Stam, *A Comparison of CEILIDH and XTR*, Algorithmic Number Theory, (ANTS 2004), LNCS 3076, (2004), 235-249.
8. D.-G. Han, T. Izu, J. Lim, and K. Sakurai, *Side Channel Cryptanalysis on XTR Public Key Cryptosystem*, IEICE Trans. Fundamentals, Special Section on Discrete Mathematics and Its Applications, VOL.E88-A, NO.5, May, pp.1214-1223, (2005).
9. D.-G. Han, T. Takagi, T.H. Kim, H.W. Kim, and K.I. Chung, *Collision Attack on XTR and a Countermeasure with a Fixed Pattern*, International Association for Cryptologic Research (IACR), Cryptology ePrint Archive 2005/316, (2005). <http://eprint.iacr.org/2005/316>
10. Kocher, C., *Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems*, Advances in Cryptology - CRYPTO '96, LNCS 1109, (1996), 104-113.
11. Kocher, C., Jaffe, J., Jun, B., *Differential Power Analysis*, Advances in Cryptology - CRYPTO '99, LNCS1666, (1999), 388-397.
12. A.K. Lenstra and E.R. Verheul, *The XTR public key system*, Advances in Cryptology - CRYPTO '00, LNCS1880, (2000), 1-19.
13. K. Okeya and T. Takagi, *The Width-w NAF Method Provides Small Memory and Fast Elliptic Scalar Multiplications Secure against Side Channel Attacks*, CT-RSA 2003, LNCS 2612, (2003), 328-342, 2003.
14. D. Page and M. Stam, *On XTR and Side-Channel Analysis*, Pre-proceedings of SAC 2004, 67-81.
15. M. Stam and A.K. Lenstra, *Speeding Up XTR*, Proceedings of Asiacrypt 2001, LNCS2248, (2001), 125-143.
16. K. Schramm, T. Wollinger, and C. Paar, *A New Class of Collision Attacks and its Application to DES*, Proceedings of FSE 2003, LNCS2887, (2003), 206-222.
17. C.D. Walter, *Simple Power Analysis of Unified Code for ECC Double and Add*, Workshop on Cryptographic Hardware and Embedded Systems 2004 (CHES 2004), LNCS 3156, (2004), 191-204.
18. R. Watro, D. Kong, S-f. Cuti, C. Gardiner, C. Lynn, and P. Kruus, *TinyPK: Securing Sensor Networks with Public Key Technology*, ACM Workshop on Security of Ad Hoc and Sensor Networks 2004 (SASN 2004), 59-64.
19. S.-M. Yen, S. Kim, S. Lim, and S. Moon, *A Countermeasure against One Physical Cryptanalysis May Benefit Another Attack*, Information Security and Cryptology 2001 (ICISC 2001), LNCS 2288, (2001), 414-427.

# Security in Persistently Reactive Systems

Takumi Endo, Junichi Miura, Koichi Nanashima,  
Shoichi Morimoto, Yuichi Goto, and Jingde Cheng

Department of Information and Computer Sciences,  
Saitama University, Saitama, 338-8570, Japan  
{endo, miura, seven, morimo, gotoh, cheng}@aise.ics.saitama-u.ac.jp

**Abstract.** From the viewpoints of dependable computing and ubiquitous computing, a new type of reactive systems, named *Persistently Reactive Systems*, was proposed. Persistently reactive systems cause some new security issues because of their continuous and persistent running without stopping their services. Based on the recognition that a persistently reactive systems can be constructed following the methodology of soft system buses, this paper defines security issues in persistently reactive systems with security requirements and security functions. To solve the issues, we propose a framework of SSB-connector, such that designers and developers can easily design and develop reliable and secure functional components of a persistently reactive system.

## 1 Introduction

From the viewpoints of dependable computing and ubiquitous computing, we have proposed a new type of reactive systems, named “*persistently reactive systems* [3, 4].” A persistently reactive system is a reactive system that functions continuously anytime without stopping its reactions even when it had some trouble, it is being attacked, or it is being maintained, upgraded, or reconfigured.

Reliability and security are matters of vital importance for any persistently reactive system to achieve persistent computing. Historically although there is a lot of work on reliability of reactive systems, few researchers focus their attention on security issues in reactive systems [1, 15, 16]. Besides, persistently reactive systems cause some new security issues because of their continuous and persistent running without stopping their services.

We consider that a persistently reactive system can be constructed as an SSB-based system [4]. Conceptually, a *soft system bus*, SSB for short, is simply a communication channel with the facilities of data/instruction transmission and preservation to connect components in a component-based system. It may consist of some *data-instruction stations*, which have the facility of data/instruction preservation, connected sequentially by *transmission channels*, both of which are implemented in software techniques, such that over the channels data/instructions can flow among data-instruction stations, and a component tapping to a data-instruction station can send data/instructions to and receive data/instructions from the data-instruction station [4]. An *SSB-based*

*system* is a component-based system consisting a group of *control components* including self-measuring, self-monitoring, and self-controlling components with general-purpose which are independent of systems, and a group of *functional components* to carry out special tasks of the system such that all components are connected by one or more SSBs and there is no direct interaction which does not invoke the SSBs between any two components [4].

We have briefly enumerated possible attacks, security requirements, and security functions for persistently reactive systems in the primitive work [7], but gave no solution to them. This paper presents refined security requirements and functions, defines security issues, and proposes a framework of SSB-connectors as a solution to them.

The rest of this paper is organized as follows: Section 2 defines security requirements of a persistently reactive system and specifies security functions and/or facilities the system must provide. Section 3 organizes security issues in persistently reactive systems. Then our solutions to the issues are proposed in Section 4. Section 5 gives a discussion about the proposed solutions from several aspects. Finally, concluding remarks are given in Section 6.

## 2 Requirements and Functions

The requirement analysis and function definition are important steps to construct any persistently reactive system with requirements of high security. A persistently reactive system built as an SSB-based system must satisfy the following security requirements.

- R1.** Since all of control components, data-instruction stations, functional components, and transmission channels may be targets of attacks, any data/instruction in the system must not be wiretapped, tempered, deleted, fabricated, reused without authenticated permission.
- R2.** Since all actions (e.g., sending and/or receiving data/instruction) in the system also may be the targets of attacks, the system must prevent attacks to any data-instruction station, any control component, and any functional component.
- R3.** Since the most intrinsic characteristic and fundamental feature of the system is continuous and persistent running without stopping its service, the system must not stop of the whole system and must provide services continuously when it is being attacked.
- R4.** For the same reason above, SSBs and control components must be able to detect any attack to the system before influence of the attack reach whole the system.
- R5.** For the same reason above, any security mechanism (e.g., key managing, protocols, or various types of algorithms) in the system must be able to be updated, exchanged, added, or deleted while running of the whole system without stopping service in case that the detect is found in those.
- R6.** Associating with above requirement, to implement computing systems suitable for ubiquitous computing, internal reconfiguration of any security

mechanism in the system must not influence providing service and must make them efficiency invisible to the user [19].

- R7.** Since SSBs and control components are general units used for any system, SSBs and control components must check the creditability of those.
- R8.** SSBs and control components must provide with a useful way to control and keep security in different degrees at different domains in the system because they intend to be used with the aim of constructing various types of computing systems.

Now then, we roughly lists only well-known security functions and/or facilities in order to satisfy the above-mentioned requirements. The intrinsic functions and/or facilities in a persistently reactive system are presented in the following sections.

- F1.** The way of encryption and decryption of data to prevent wiretapping.
- F2.** The way of hash value or message authentication code (MAC) to detect tempering.
- F3.** The way of digital signature to detect tampering and/or fabricating.
- F4.** The way of nonce (e.g., time stamp or random number) to prevent replaying.
- F5.** The way of setting up expiration of a data/instruction to prevent reusing those.
- F6.** The way of authorization or combination of access control and authentication to prevent unauthenticated access.

In order to satisfy the requirements and to provide functions defined in this section, security issues to be addressed in persistently reactive systems are clearly defined in the following section.

### 3 Security Issues

Building upon the foundations presented in the previous section, we now investigate the security issues in a persistently reactive system built as an SSB-based system.

Security issues in a persistently reactive system can be classified into two types, i.e., general issues for any persistently reactive system and system-dependent ones. This paper focuses attention on the former issues and does not mention the latter ones, because SSBs and control components never indicate to system designers and developers about the system-dependent issues: how to design and develop a 'secure' functional component.

The general issues are related to control components, SSBs (data-instruction stations and transmission channels), and interface between a functional component and an SSB. We define the general issues as follows:

1. **Dynamic Reconfiguration and Evolution of Security Mechanism.**  
Almost all security mechanisms cannot proof and verify that it is secure 'for

eternity'. Therefore, to implement a persistently reactive system, any security mechanism must be able to be reconfigured (updated, exchanged, added, or deleted) while running of the whole system without stopping service. In other words, persistently reactive systems must have evolutionary security.

2. **Three Types of Security for Versatility.** SSBs and control components intend to be used with the aim of constructing various types of reactive systems. It is well known that there is a trade-off between the security and the efficiency/performance of any system. Is it precious to pay a price of efficiency for getting high security? This class of security issues can be subdivided as follows:
  - (a) **Multilevel Security.** For example, a real-time system, a special reactive system with requirements of real-time processing, would not demand passing the time by calculating with security (e.g., encryption/decryption). Accordingly, SSBs and control components must be able to deal with various security policies of a system efficiently.
  - (b) **Multilateral Security.** For similar reason with the previous one we can say multilateral security, that is, there are multiple 'local' security policies in one system. In a large-scale system or a case of integrating multiple running systems, there are various domains to control and keep security in each degree. SSBs and control components need to be able to deal with the concept of various security policies in one system efficiently.
  - (c) **Security with Various Importance Degree in a Data/Instruction.** Data/instructions, used for communicating between any two components, can be classified into control data and functional data. The control data are those data that flow from a control component or to a control component, the functional data are those data exchanged between functional components. More realistically, there will be various importance degree of data/instruction in a persistently reactive systems even in the case of communication between components belong to same domain in which security policies are shared. How SSBs and control components distinguish these various importance degree of data/instructions?
3. **Complexity.** Many security defects are resulted from the complexity in computing systems. A motivated idea of the SSB methodology is Albert Einstein's proverb: "Everything should be made as simple as possible, but not simpler," that is, the basic idea underling the SSB methodology is to control the complexity of information processing in a target system by making the structure simplicity of the system [4]. Can the solutions to the above-mentioned issues really keep down or reduce the complexity of design, development, and maintenance of persistently reactive systems? A solution to the issue No. 1, reconfigurable and evolutionary security, must not lead to enveloping some older defects or weaknesses by use of newer ones.
4. **'Running/Serving Stop' Attacks.** Among the attacks a persistently reactive systems may face, the most serious one should be 'running/serving stop' attacks, just like 'running/serving stop' errors in testing and debugging persistently reactive systems [3]. This is due to that the most essential and/or general requirement for persistently reactive systems is continuous

and persistent running without stopping services. Now then, the simplest way to stop running/serving of the systems would be attacking the control components. Consequently, we face the next issue in association with this class of attacks.

5. **Protecting Control Components.** Obviously, necessary condition and/or fundamental assumption to promise the security of a persistently reactive system is that the control components assure and keep the highly security to themselves. Constructing such a ‘perfect’ control components, however, is probably impossible as well as constructing such computing systems (cf. issue No. 1). In addition, since control components cannot be maintained and upgraded in the same manner as them of functional components [4, 5], protecting the control components is much more difficult than protecting a functional component. We need to find new countermeasures.

This paper mainly focuses on how to solve the issues No. 1, 2, and 3 and presents the solution to these three issues in the next section.

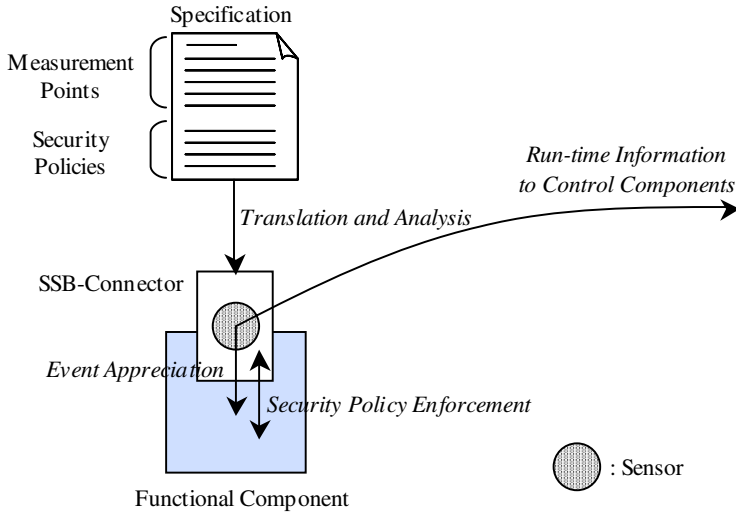
## 4 A Framework of SSB-Connectors

In this section we propose the notion of *SSB-connectors*. An SSB-connector is embedded among each functional component, and acts multiple common roles among functional components in a persistently reactive system to support design and development of the functional component.

### 4.1 Specifications to Describe Measurement Points and Security Policies

Figure 1 illustrates a framework of SSB-connector with specification to describe measurement points and security policies. At first, an SSB-connector translates a specification. The specification, described by the system designers and/or developers, is written in a specification language and conformed to some rules. And then the SSB-connector analyzes measurement points and security policies. Thereby, a sensor is generated mechanically. A sensor is statements which perceive the events in a functional component and send run-time information to control components. The SSB-connector also enforces the security policies based on the analytical result of security polices. The specifications can be rewritten while the system running. If the specifications are rewritten, the SSB-connector recompile them.

The basic idea underlying SSB-connectors is applying the supporting tool proposed by Nonaka et al. [12, 13] to the SSB methodology. Their supporting tool understand the *measurement specifications* for Ada programs and support systematic development of concurrent systems based on the self-measurement principle [2]. In the measurement specifications, *measurement points* are described [12, 13]. A measurement point is a location in a *target program* where an event concerning some attribute of an object specified by measuring and monitoring requirements occurs during an execution of the program. A target program is



**Fig. 1.** A Framework of SSB-connector with Specification to Describe Measurement Points and Security Policies

a set of program source texts that implemented the functional components [2]. In our specifications, the security policies in addition to the measurement points are described.

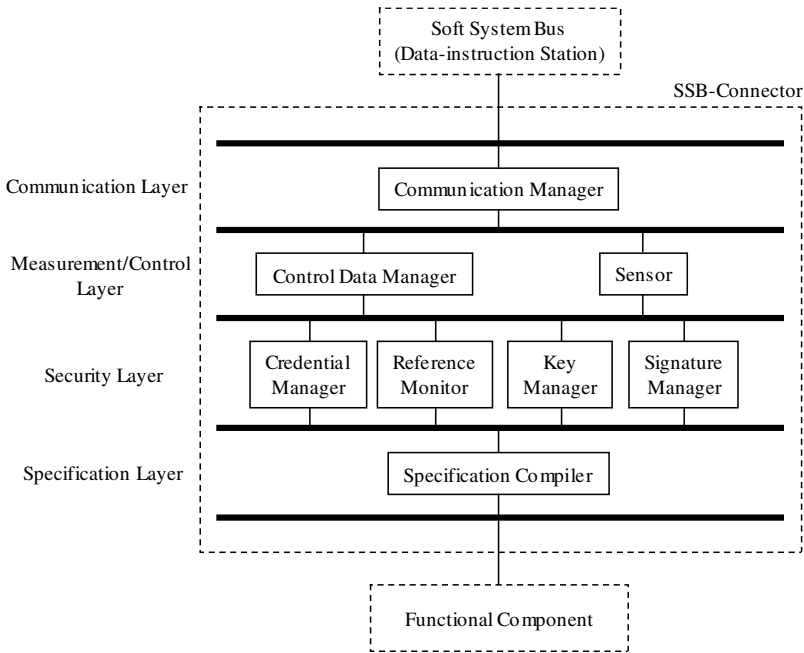
Using the SSB-connector and the specifications, designers and developers can more easily design and develop a functional component. A functional component must consist of functional part of the component and non-functional part for the reliability and security which SSBs and control component promise. These two parts are clearly separated by use of SSB-connectors. Designers and developers just need to design and develop functional part, and they just need to describe specifications about non-functional part.

#### 4.2 Hierarchical and Alternative Structure in SSB-Connectors

Our solution for enabling reconfiguration and evolution of security mechanisms used in a persistently reactive system, is to hierarchize the SSB-connector and subdividing it into some reconfigurable blocks. Figure 2 roughly shows an internal architecture in an SSB-connector. The internal architecture in an SSB-connector has four layers and consist of eight blocks.

The *communication layer* handles the interaction of a functional component with others. This layer has the *communication manager*, which functions abstraction of connection protocols and conversion of real address and virtual address.

The *measurement/control layer* acts the role of intermediate medium between functional components and control components abstractly. This layer has following two main blocks. The *control data manager* stores control data in the com-



**Fig. 2.** Internal Architecture in an SSB-connector

ponent. The *sensor* perceives the events in the component and create control data for sending control components.

The *security layer* handles various security policies enforcement based on the analytical result of compiling the specification. This layer has following four main blocks. The *credential manager* is responsible for maintaining the locally cached identity information. The *reference monitor* identifies the access authorities of entities or intercept of perilous events to the component. The *key manager* generates the unique key pairs and store those. The *signature manager* is responsible for signing requests and verifying notifications.

The *specification layer* handles translation of the specifications. This layer has the *specification compiler*, translates the specification, and generate sensor or control data based on the analysis result.

Each blocks in an SSB-connector must provide some reconfigurable security mechanisms. Therefore, it is also desired that there are multiple alternative in a block. The advantage of subdividing into some appendicle blocks and preparing spare mechanisms is that the influence of reconfiguration in one block is lightly affected. Besides, if these security mechanisms can be upgraded, exchanged, added, or deleted while other units and/or components running, it will mean evolutionary security.

The structure sketched in this section is primitive. Therefore, the structure might overslip some necessary units, or these units might be divided more-minutely. For example, the specification compiler would be subdivided into syn-



tax analyzer, semantic analyzer, sensor-code generator, and converter of the security policies, and this subdivision might be better than packing as a one unit. Further investigation and design of better (ideally, the best) structures of SSB-connectors are needed.

## 5 Discussion

At the beginning, we summarize how the security issues No. 1, 2, and 3 defined in Section 4 can be solved in SSB-connectors. First, proposed SSB-connectors are framework which can be reconfigured some alternatives dynamically based on the recognition that it is impossible to prepare the undefeatable mechanisms previously. Key features to enable suchlike dynamic reconfiguration are to hierarchize the structures of the connectors and to divide them into reconfigurable blocks. If control components can measure, monitor, and control the whole system and manage the connectors under such structures, dynamic reconfiguration and evolution of security mechanism will be possible. Second, if SSB-connectors can translate and analyze various security policies, specified by designers and developers of each components, and the four blocks in the security layer can handle data or accesses, multilevel and multilateral security issues can be resolved. And to manage various importance degree of data/instructions, communication manager communicate with other components leveraging the processing results of credential manager, key manager, and signature manager in the security layer. In this regard, appending appropriate information to data/instructions and valid communication processes must be needed. Third, the complexity of security design and development must be decreased by separating non-functional parts from component development process as a SSB-connector, following the systematical and unified system design and development methodology based on soft system buses.

The essential sense of our work is to construct such the system design and development methodology, and SSB-connectors are expected to support the design and development of functional components in the methodology. As related works to SSB-connectors, a hierarchical security structure such like PACE [18] or an approach of security wrappers [8] by the automated run-time enforcement of security policies in a component-based system are established. The major difference between our work on SSB-connectors and these similar works is that our work take the intrinsic requirements of continuous and persistent running into account but there is no such consideration and requirement on the related works.

The work in this paper is still in progress and as such there remains many implementation issues to be addressed. The major future and ongoing works are as follows:

1. We must closely investigate whether the supporting tool of measurement specifications [12, 13] and security policy enforcement [10], both of which are shown its implementability in some systems (not persistently reactive systems), can be applied to the SSB methodology. If it is not possible to apply, we must need the development of some new technologies or modification of these works.

2. We must design a specification language with measurement points and security policies and their specification rules. Till now, there are many formal specification language (e.g., Z, VDM), including security policy specification languages such as Ponder [17] or specification languages for concurrent systems such as CCS [11]. However, to our knowledge, no language can deal with both measurement points and security policies efficiently.
3. We must work on many case studies to demonstrate the availability of the SSB-methodology and proposed SSB-connectors, considering various configuration, for example, the number of SSBs in a system, structure of an SSB (linear or circular), information flow direction along an SSB (one-way or bidirectional) , or how components are connected to SSBs.

We lastly present an idea for increasing the capabilities of proposed SSB-connectors. The idea is use of patterns. Design patterns are known as well-established approach to solve complexity and to improve productivity. We think that measurement points and security policies in specifications can be made some patterns. If it is possible to make some patterns, control components would provide the facility like repository, store multiple measurement patterns or security policies. And by picking up necessary or valuable patterns from among them in design and development of a functional component, designers and developers would be design and develop of the component more easily and productively.

## 6 Concluding Remarks

We have specified security requirements and functions in persistently reactive systems built following the methodology of soft system bus. We also defined security issues and proposed the framework of SSB-connectors with specifications in order to solve the issues. Moreover, adequacy and implementation issues of this idea are discussed.

We believe the investigation and design of SSB-connectors sketched in this paper are good starting points for supporting development for persistently reactive systems, but plenty of work remains to be done. A remaining big (or maybe the biggest) technical challenge in persistently reactive systems is how to protect, maintain, upgrade, reconfigure control components [5]. What are countermeasures against unexpected attacks to control components? We think that we may be able to get some important hints from autonomic computing [9, 6] and recovery-oriented computing [14].

## References

1. Backes, M., Pfitzmann, B., Waidner, M.: A General Composition Theorem for Secure Reactive Systems. In Naor, M. (ed.): TCC 2004. Lecture Notes in Computer Science, Vol. 2951, Springer-Verlag (2004) 336–354
2. Cheng, J.: The Self-Measurement Principle: A Design Principle for Large-scale, Long-lived, and Highly Reliable Concurrent Systems. In: Proc. 1999 IEEE International Conference on Systems, Man and Cybernetics, Vol. 4. (1998) 4010–4015

3. Cheng, J.: Testing and Debugging Persistently Reactive Systems - A New Challenge in Software Engineering. In: Proc. Japan Symposium on Software Testing 2005. (2005) 34–40
4. Cheng, J.: Connecting Components with Soft System Buses: A New Methodology for Design, Development, and Maintenance of Reconfigurable, Ubiquitous, and Persistent Reactive Systems. In: Proc. 19th International Conference on Advanced Information Networking and Applications, Vol. 1. (2005) 667–672
5. Cheng, J.: Comparing Persistent Computing with Autonomic Computing. In: Proc. 11th IEEE-CS International Conference on Parallel and Distributed Systems, Volume II Workshops. (2005) 428–432
6. Chess, D.M., Palmer, C.C., White, S.R.: Security in an Autonomic Computing Environment. *IBM Systems Journal* **42** (2003) 107–118
7. Endo, T., Miura, J., Nanashima, K., Morimoto, S., Goto, Y., Cheng, J.: Security Issues in Persistently Reactive Systems (fast abstract). In: Supplemental Volume of the 2005 International Conference on Dependable Systems and Networks. (2005) 56–57
8. Herrmann, P., Wiebusch, L., Krumm, H.: State-based Security Policy Enforcement in Component-based E-commerce. In: Proc. 2nd IFIP Conference on E-Commerce, E-Business, and E-Government (I3E), Kluwer Academic Publisher (2002) 195–209
9. Kephart, J.O., Chess, D.M.: The Vision of Autonomic Computing. *IEEE Computer* **36** (2003) 41–50
10. Kühnhauser, W.E.: A Paradigm for User-defined Security Policies. In: Proc. 14th IEEE Symposium on Reliable Distributed Systems. (1995) 135–144
11. Milner, R.: *Communicating and Mobile Systems: The  $\pi$ -Calculus*. Cambridge University Press (1999)
12. Nonaka, Y., Cheng, J., Ushijima, K.: A Supporting Tool for Development of Self-measurement Ada Programs. In Keller, H.B., Plodereder, E., eds.: *Reliable Software Technologies – Ada-Europe 2000*. Lecture Notes in Computer Science, Vol. 1845, Springer-Verlag (2000) 69–81
13. Nonaka, Y., Cheng, J., Ushijima, K.: Measurement Specifications and Their Applications for Development of Concurrent Self-Measurement Programs. *IPSJ Journal* **43** (2002) 743–753 (in Japanese with English summary).
14. Patterson, D. et al.: *Recovery Oriented Computing (ROC): Motivation, Definition, Techniques, and Case Studies*. Technical report, UC Berkeley Computer Science UCB//CSD-02-1175 (2002)
15. Pfizmann, B., Schunter, M., Waidner, M.: Cryptographic Security of Reactive Systems. *Electronic Notes in Theoretical Computer Science* **32** (2000)
16. Pfizmann, B., Waidner, M.: Composition and Integrity Preservation of Secure Reactive Systems. In: Proc. 7th ACM Conference on Computer and Communications Security. (2000) 245–254
17. Policy Research Group: *Ponder: A Policy Language for Distributed Systems Management* <http://www-dse.doc.ic.ac.uk/Research/policies/ponder.shtml>.
18. Suryanarayana, G., Erenkrantz, J.R., Hendrickson, S.A., Taylor, R.N.: PACE: An Architectural Style for Trust Management in Decentralized Applications. In: Proc. 4th Working IEEE/IFIP Conference on Software Architecture. (2004) 221–232
19. Weiser, M.: The Computer for the 21st Century. *Scientific American* **265** (1991) 94–104

# ID-Based Access Control and Authority Delegations

So-Young Park and Sang-Ho Lee

Dept. of Computer Science and Engineering, Ewha Womans University,  
11-1 Daehyun-Dong, Seodaemun-Gu, Seoul, Korea  
soyoung@ewhain.net, shlee@ewha.ac.kr

**Abstract.** In ubiquitous computing environments, a user can utilize a certain service in anywhere and anytime using any devices, then authentication and authorization are crucial for security in pervasive computing. In this paper, we propose a new ID-based access control model adequate for ubiquitous computing. We eliminate the use of X.509 certificates of trusted parties for authentication and allow ID-based authority delegations for flexible access control in distributed networks. We implement our proposed model using a bilinear map such as Weil pairing, and we prove the security of the proposed model against chosen-message attack under the random oracle model.

## 1 Introduction

### 1.1 Background and Related Work

With our computing environments becoming ubiquitous, it is able to be realized to utilize any services what a user wants in anywhere and anytime using any devices. In order to be possible seamless and secure communications under distributed and heterogeneous network environments, in addition to confidentiality, integrity and availability that are usually primary goals of any security systems, authentication and authorization are crucial for security in pervasive computing. Due to the distributed nature of networks, we cannot assume an architecture with a central authority for authentication such like Kerberos[15]. Distributed trust can be a solution to this problem. However, the well-known existing security infrastructure[7][8][12][14][16][2] that deal with authentication and authorization in distributed networks are based on X.509[17] certificates in public key infrastructure (PKI). Certificate-based authentication require very high computational costs and bandwidth, moreover they cause other certificate-related management problems including revocation, storage and distribution[10]. Kagal et al.[11] proposed a distributed trust model for a secure Smart Office not with certificates but with XML signatures from a trusted authority.

We suggest a flexible access control model adequate for pervasive computing. Let us consider the following ubiquitous service to illustrate our potential application. There is a service provider that provides multimedia services, and a client Alice registered to the service provider with her identity. Alice has several

types of devices such as a personal computer, a cellular phone, a PDA or a laptop, etc., and wants to enjoy a certain service on different devices continuously regardless to her location. However, Alice does not want to register all her devices to the service provider. Then, how can the service provider authenticate the authorization of each device? We provide a lightweight ID-based access control model. Each device has its own identity such as an IP address or a phone number, etc. The device can create a token for authentication using its identity with the help of the device holder and send it to the service provider. Then, the service provider can authenticate the device by verifying the token using Alice's identity. In addition, Alice wants to enjoy the same multimedia services on TV which is in her friend's house with her friend John, or on some other foreign devices. But Alice does not want her personal or secure information to be written or memorized in the foreign devices. Or, John wants to enjoy a certain multimedia service of the service provider on his TV for a short period of time under the agreement of Alice, but he does not want to register to the service provider. In this case, Alice can delegate her access authority to John's device temporarily. Alice creates a delegation token on the identity of John's device using her own device (for example, a cellular phone) and sends it to John's device. Then, John tries to access the service with the delegation token, and the service provider can decide whether John's device is authorized or not, using the identity of the device and the delegation token.

## 1.2 Our Contribution

In this paper, we introduce a new ID-based access control model appropriate to ubiquitous computing. We eliminate the use of X.509 certificates, and allow ID-based authority delegations.

***Decentralized-Certificateless Authentication:*** We do not assume a centralized authentication server such as Kerberos or the uses of certificates of a trusted authority for authentication. We assume that the entire network is decentralized and collections of several different types of sub-networks. We remove the need for certificates and some of the problems associated with them by applying ID-based cryptography[13][3][4][6] to our scheme. Using the Gap Diffie-Hellman property of the pairing, we implement a lightweight ID-based access control model without certificates.

***ID-Based Authority Delegation:*** We provide ID-based authority delegations. A delegation is a temporary permit issued by the user and given to another subject that authorizes the subject to act on the user's behalf[5]. We allow authority delegations to be carried out between two subjects in different network domains using the identities of the subjects without any exposure of private or secure information of the subjects, and without any helps (authorization) of a trusted party. We assume a temporary delegation like one-time permission. The delegated subject can use a certain service for a short period of time, and once the delegation expires, the delegated subject is denied access to any services.

Our ID-based access control model is built from a modified bilinear map such as the weil pairing on elliptic curves[3]. We define a chosen-message security for our ID-based access model and analyze the security of the proposed scheme against the chosen-message attack under the random oracle model.

The rest of the paper is organized as follows. In Section 2, we review the notion of a bilinear map. We formalize our model and explain concrete descriptions of ID-based access control scheme in Section 3. In Section 4, we analyze the security of the proposed scheme, and then conclude the paper in Section 5.

## 2 Bilinear Map

We briefly review the bilinear map and several hard problems which provide the basic security of our scheme. Boneh and Franklin introduced a bilinear map called a "pairing" in their IBE scheme. Typically, the pairing used is a modified Weil or Tate pairing on a supersingular curve or abelian variety[3][1]. Let  $G_1$  be an additive group of prime order  $q$  and  $G_2$  be a multiplicative group of the same order. Our scheme makes use of an admissible bilinear map  $\hat{e} : G_1 \times G_1 \rightarrow G_2$  between these two groups. The map must satisfy the following properties:

1. Bilinear: We say that a map  $\hat{e} : G_1 \times G_1 \rightarrow G_2$  is bilinear, for all  $Q, W, Z \in G_1$ , if  $\hat{e}(Q, W + Z) = \hat{e}(Q, W)\hat{e}(Q, Z)$  and  $\hat{e}(Q + W, Z) = \hat{e}(Q, Z)\hat{e}(W, Z)$ . Consequently, for any  $a, b \in Z_q^*$ ,  $\hat{e}(aQ, bW) = \hat{e}(Q, W)^{ab} = \hat{e}(abQ, W)$ .
2. Non-degenerate: The map does not send all pairs in  $G_1 \times G_1$  to the identity in  $G_2$ . Observe that since  $G_1, G_2$  are prime order groups, it is implied that if  $P$  is a generator of  $G_1$  then  $\hat{e}(P, P)$  is a generator of  $G_2$ .
3. Computable: There is an efficient algorithm to compute  $\hat{e}(Q, W)$  for all  $Q, W \in G_1$ .

In general,  $G_1$  is a cyclic subgroup of the additive group of points on a supersingular elliptic curve over  $E(F_p)$ .  $G_2$  is a cyclic subgroup of the multiplicative group associated with a finite extension of  $F_p$ .

The security of the pairing-based schemes relies on the hardness of the following problems.

- Discrete Logarithm Problem (DLP): Given two group elements  $P$  and  $Q = nP$  in  $G_1$ , find  $n$ .
- Computational Diffie-Hellman Problem (CDHP): For any  $a, b \in Z_q^*$ , given  $\langle P, aP, bP \rangle$ , compute  $abP$ .
- Decisional Diffie-Hellman Problem (DDHP): For any  $a, b, c \in Z_q^*$ , given  $\langle P, aP, bP, cP \rangle$ , decide whether  $c \equiv ab \pmod q$ . DDHP in  $G_1$  is easy. To see this, observe that given  $\langle P, aP, bP, cP \rangle \in G_1$  we have

$$c = ab \pmod q \Leftrightarrow \hat{e}(P, cP) = \hat{e}(aP, bP)$$

- Gap Diffie-Hellman Problem (GDHP): A class of problems where DDHP is easy while CDHP is hard.
- Bilinear Diffie-Hellman Problem (BDHP): For any  $a, b, c \in Z_q^*$ , given  $\langle P, aP, bP, cP \rangle$ , compute  $\hat{e}(P, P)^{abc} \in G_2$ .

### 3 ID-Based Access Control Scheme

In this section, we present a formal definition for our ID-based access control scheme, and then we explain concrete descriptions of the scheme.

#### 3.1 System Model and Security Requirements

Our ID-based access control model basically consists of a service provider, users (or subjects) and devices to be able to utilize certain services of the service provider. We assume that each user has  $n$ -types of devices, and that all devices have their own identities such as IP addresses or phone numbers, etc., and can communicate each other through wired or wireless networks.

**Definition 1.** *We define that access control is a mechanism that monitors and controls the subjects access to the service provider's resources. The term of control means to decide to assign access rights to the subjects after authenticating whether the subjects are authorized or not.*

A detailed system model for ID-based access control is described below. Let  $k$  be a security parameter, and let  $\mathcal{IG}$  be a BDH parameter generator.

**Definition 2.** *An ID-based access model is specified by six algorithms ( $Gen_{Sys}$ ,  $Reg$ ,  $GEN_{AToken}$ ,  $GEN_{DToken}$ ,  $Auth_{Device}$ ,  $Auth_{FDevice}$ ) such that:*

1.  $Gen_{Sys}$ : *It takes as input the security parameter  $k$  and  $\mathcal{IG}$ . It returns system parameters.*
2.  $Reg$ : *It takes as input a user  $A$ 's identity  $ID_A$ ,  $Auth-ID_A$ , user information and service-related information. Then, it adds the information to Access Control List (ACL) maintained by the service provider.*
3.  $GEN_{AToken}$ : *It takes as input  $A$ 's secure master identity  $MID_A$ , a device identity  $DID$  and an access-request message. It generates an authentication token for the device.*
4.  $GEN_{DToken}$ : *It takes as input  $A$ 's secure master identity  $MID_A$ , a device identity  $DID$  and an authority-delegation message. It generates a delegation token for the device.*
5.  $Auth_{Device}$ : *It takes as input an access-request message, a device identity and an authentication token. It decides to assign an access right to the device or not.*
6.  $Auth_{FDevice}$ : *It takes as input an access-request message, a authority-delegation message, a device identity and a delegation token. It decides to assign an access right to the device or not.*

Our goal is to provide a secure access control process and a secure delegation process on "open" insecure networks. Since we assumed that the network is insecure, an adversary can intercept and modify all data transmitted on the network, and also can retransmit a valid data transmission intercepted by the adversary maliciously or fraudulently (replay attack). In addition, we assume that the adversary can do chosen-message attacks[9]. We will formalize the chosen-message

security model of our scheme against the chosen-message attacks in Section 4. Under the above attack environments, we consider the proposed scheme as satisfying at least the following security requirements for secure access control and delegations.

- **Authorization:** An access controller of a service provider can authenticate whether each subject is authorized or not.
- **Unforgeability:** The adversary cannot forge a valid token by an authorized subject without knowing secure information of the subject.
- **Accountability:** An authorized subject that carried out a delegation cannot repudiate the fact of that the subject delegated its authority to a foreign subject.
- **Revoking:** The delegator must have the ability to cancel delegations it has issued (before the valid date of delegation is expired).
- **Replay Attack Resistance:** The access controller can detect and protect the replay attack by the adversary or any subjects.

### 3.2 Implementation of the ID-Based Access Control Scheme

We describe the six algorithms needed to define our ID-based access control scheme. For a simple explanation, let a service provider be  $SP$ , a user be  $A$ , the user's device be  $DA_i$ , a foreign user be  $B$  and the foreign user's device be  $DB_i$ , where  $i = 1, \dots, n$ . First, the  $SP$  generates its system parameters as follows:

$Gen_{Sys}$ : The  $SP$

1. runs  $\mathcal{IG}$  on input  $k$  to generate groups  $G_1, G_2$  of some prime order  $q$  and a bilinear map  $\hat{e} : G_1 \times G_1 \rightarrow G_2$ ;
2. chooses arbitrary generators  $P \in G_1$ , and chooses cryptographic hash functions  $H_1 : \{0, 1\}^* \rightarrow G_1, H_2 : \{0, 1\}^* \rightarrow Z_q^*$ .

The system parameters are  $params = (G_1, G_2, \hat{e}, P, H_1, H_2)$ , and those are publicly available for its clients. Now, we suppose that  $A$  wants to register to the  $SP$ . The algorithm of  $Reg$  is carried out between  $A$  and  $SP$ .

$Reg$ :

1. The  $SP$  sends the system parameters to  $A$ .
2.  $A$  makes its master secure identity  $MID_A \in \{0, 1\}^*$ ;
3. and sets  $Auth-ID_A = H_2(MID_A) \cdot P \in G_1$ ;
4. and sends  $\langle ID_A, Auth-ID_A \rangle$  with other user information required to registration to  $SP$ .
5.  $SP$  adds the user information to its access control list.

$SP$  maintains an access control list (ACL) for client authentication. We suppose that ACL basically consists of the attributes of client  $ID$ ,  $Auth-ID$ , user information, contents of services and service-related information. User information can contain some private personal information that the  $SP$  wants to know



including full-name, age, mail address and phone number, etc. The contents of services can contain services allowed to  $A$ , and the service-related information can contain constraints or policies defined by  $SP$  about the services. The important thing is that  $SP$  does not contain any secret information of its clients such as passwords.  $SP$  can authenticate its clients using the pair of  $(ID, Auth-ID)$  in ACL, and can provide the corresponding services to clients.

**Authority Authentication.** We suppose that  $A$  wants to access to  $SP$  using her own device  $DA_i$ . Then  $A$  performs  $GEN_{AToken}$  on  $DA_i$  as follows:

$GEN_{AToken}$ :

1.  $A$  inputs  $ID_A$  and  $MID_A$  to  $DA_i$ .
2.  $DA_i$  generates  $access\text{-}req = 'access\text{-}request' || ID_A || DID_{DA_i} || Time_i$ , where  $DID_{DA_i}$  is the identity of  $DA_i$ , and  $Time_i$  is current time information such like GMT.
3.  $DA_i$  computes  $a = \frac{H_2(MID_A)}{H_2(DID_{DA_i})} \in Z_q^*$  and  $AT_A = a \cdot H_1(access\text{-}req) \in G_1$ .

When  $SP$  receives an access signal from a certain device  $DA_i$ ,  $SP$  carries out  $Auth_{Device}$  to authenticate  $DA_i$  as follows:

$Auth_{Device}$ :

1.  $DA_i$  sends  $\langle access\text{-}req, AT_A \rangle$  to  $SP$ .
2.  $SP$  checks  $access\text{-}req$  and gains  $Auth-ID_A$  from ACL corresponding to  $ID_A$ .
3.  $SP$  computes  $H_1(access\text{-}req) \in G_1$  and  $H_2(DID_{DA_i}) \in Z_q^*$ .
4.  $SP$  checks if  $\hat{e}(AT_A, H_2(DID_{DA_i}) \cdot P) = \hat{e}(H_1(access\text{-}req), Auth-ID_A)$ .
5. If the equation holds,  $SP$  assigns an access right to  $DA_i$ .

**Authority Delegations.** An authorized subject can delegate its authority to a foreign subject that the authorized subject trusts. We suppose  $A$  wants to delegate her authority to a foreign device  $DB_i$  that  $B$  holds. We assume that  $A$  has her own device  $DA_i$ .  $A$  and  $B$  carry out  $GEN_{DToken}$  as follows:

$GEN_{DToken}$ :

1.  $B$  inputs  $ID_B$  to  $DB_i$ .
2.  $DB_i$  generates  $auth\text{-}deleg\text{-}req = 'authority\text{-}delegation\text{-}request' || ID_B || DID_{DB_i} || Time_i$ , where  $DID_{DB_i}$  is the identity of  $DB_i$  and  $Time_i$  is current time information.
3.  $B$  sends  $auth\text{-}deleg\text{-}req$  to  $DA_i$ .
4.  $A$  inputs  $ID_A$  and  $MID_A$  to  $DA_i$ .
5.  $DA_i$  generates  $auth\text{-}deleg\text{-}msg = 'authority\text{-}delegation\text{-}message' || ID_A || DID_{DB_i} || Contents\ of\ Services || Time_i$ , where 'Contents of Services' are a list of services allowed to  $DB_i$ .
6.  $DA_i$  computes  $d = \frac{H_2(MID_A)}{H_2(DID_{DB_i})} \in Z_q^*$  and  $DT_A = d \cdot H_1(auth\text{-}deleg\text{-}msg) \in G_1$ , and sends  $\langle auth\text{-}deleg\text{-}msg, DT_A \rangle$  to  $DB_i$ .

When  $SP$  receives an access signal from a certain device  $DB_i$ ,  $SP$  carries out  $Auth_{FDevice}$  to authenticate  $DB_i$  as follows:

*Auth<sub>FD</sub>Device*:

1.  $DB_i$  generates  $\text{deleg-access-req} = \text{'delegate-access-request'} || \text{DID}_{DB_i} || \text{Time}_i$ .
2.  $DB_i$  sends  $\langle \text{deleg-access-req}, \text{auth-deleg-msg}, DT_A \rangle$  to  $SP$ .
3.  $SP$  checks  $\text{deleg-access-req}$  and  $\text{auth-deleg-msg}$ , and gains  $\text{Auth-ID}_A$  from ACL corresponding to  $ID_A$ .
4.  $SP$  computes  $H_1(\text{auth-deleg-msg}) \in G_1$  and  $H_2(\text{DID}_{DB_i}) \in Z_q^*$ .
5.  $SP$  checks if  $\hat{e}(DT_A, H_2(\text{DID}_{DB_i}) \cdot P) = \hat{e}(H_1(\text{auth-deleg-msg}), \text{Auth-ID}_A)$ .
6. If the equation holds,  $SP$  assigns an access right to  $DB_i$ .

## 4 Security Analysis

In this section, we analyze the security of our proposed scheme about the requirements mentioned in Section 3.1.

**Authorization:** The service provider assigns an access right to subjects with tokens created by master identities corresponding to *Auth-IDs* pre-registered to the service provider. If the tokens are valid, the service provider can authenticate that the subjects with the tokens are authorized, because only users who know the master identities corresponding to *Auth-IDs* can create the valid tokens.

**Unforgeability:** We show that the tokens cannot be forged by adversaries at below. We prove the security of our scheme against existential forgery under a chosen-message attack in the random oracle model. Existential forgery means that the adversary attempt to forge the valid tokens, on messages of his choice, by an authorized subject. The token generation protocols of our scheme are similar to create a kind of short signature from Weil pairing[4]. Therefore, our security analysis follows basically the security proof in Boneh et al's scheme[4] except that our system parameters are selected in two groups  $G_1, G_2$  where  $G_1$  is an additive group and  $G_2$  is a multiplicative group, with the same prime order  $q$ .

At first, we formalize a chosen-message security model. In this model, the adversary  $\mathcal{A}$  is given public  $ID$  and  $\text{Auth-ID}$ . His goal is to existential forgery of a token. We give the adversary power to choose all public  $ID$ s and  $\text{Auth-ID}$ s except the challenge public  $ID$  and  $\text{Auth-ID}$ . The adversary is also given access to a token generating oracle on the challenge's master  $ID$ . His advantage  $\text{AdvToken}_{\mathcal{A}}$ , is defined to be his probability of success in the following game.

- **Setup:** The adversary  $\mathcal{A}$  is provided with a public  $ID_{\mathcal{A}}$  and  $\text{Auth-ID}_{\mathcal{A}}$ , generated at random.
- **Queries:** Proceeding adaptively,  $\mathcal{A}$  requests tokens with  $\text{Auth-ID}_{\mathcal{A}}$  on messages of his choice.
- **Response:** Finally,  $\mathcal{A}$  outputs a message  $M_{\mathcal{A}}$ , and a token  $\text{Token}_{\mathcal{A}}$  on  $M_{\mathcal{A}}$ .

The adversary wins if the token  $\text{Token}_{\mathcal{A}}$  is a valid token on message  $M_{\mathcal{A}}$  under  $\text{Auth-ID}_{\mathcal{A}}$ , and  $\text{Token}_{\mathcal{A}}$  is non-trivial.

**Definition 3.** An adversary  $\mathcal{A}(t, q_H, q_T, \epsilon)$ -breaks a token generating scheme in the chosen-message security model if:  $\mathcal{A}$  runs in time at most  $t$ ;  $\mathcal{A}$  makes at most  $q_H$  queries to the hash function and at most  $q_T$  queries to the token generating oracle;  $\text{AdvToken}_{\mathcal{A}}$  is at least  $\epsilon$ . A token generating scheme is  $(t, g_H, q_T, \epsilon)$ -secure against existential forgery in the chosen-message model if no adversary  $(t, g_H, q_T, \epsilon)$ -breaks it.

**Lemma 1.** Let  $(G_1, G_2)$  be a  $(t', \epsilon')$ -GDH group pair of order  $q$ . Then the token generation scheme on  $(G_1, G_2)$  is  $(t, q_H, q_T, \epsilon)$ -secure against existential forgery under an adaptive chosen-message attack for all  $t$  and  $\epsilon$  satisfying  $\epsilon \geq e(q_T + 1) \cdot \epsilon'$  and  $t \leq t' - c_{G_1}(q_H + 2q_T)$ , where  $c_{G_1}$  is multiplication time in  $G_1$  for each query, and  $e$  is the base of the natural logarithm.

**Proof.** Suppose  $\mathcal{A}$  is a forger algorithm that  $(t, g_H, q_T, \epsilon)$ -breaks the token generating scheme. We show how to construct a  $t'$ -algorithm  $\mathcal{B}$  that solves CDH problem with probability at least  $\epsilon'$ . This will contradict the fact that  $G_1$  is a  $t', \epsilon'$ -CDH group. Let  $P$  be a generator of  $G_1$ . Algorithm  $\mathcal{B}$  is given  $P, A\text{-ID}, Q \in G_1$  and a random  $z \in Z_q^*$ , where  $A\text{-ID} = a \cdot P$  for a random secure  $a \in Z_q^*$ . Its goal is to output  $(a/z) \cdot Q \in G_1$ . Algorithm  $\mathcal{B}$  simulates the challenger and interacts with the forger  $\mathcal{A}$  as follows:

**Setup:** Algorithm  $\mathcal{B}$  starts by giving  $\mathcal{A}$  the generator  $P$  and the public  $\text{Auth-ID}_1 = A\text{-ID} + r \cdot P = a \cdot P + r \cdot P$  where  $r$  is a random value in  $Z_q^*$ .

**H-queries:** At any time algorithm  $\mathcal{A}$  can query the random oracles of  $H$ . To respond to these queries algorithm  $\mathcal{B}$  maintains a list of tuples  $(ID_j, z_j, M_j, w_j, b_j, c_j)$  as explained below. We refer to this list as the  $H$ -list. The list is initially empty. When  $\mathcal{A}$  queries the oracle  $H$  at points of  $ID_i$  and  $M_i \in \{0, 1\}^*$ , algorithm  $\mathcal{B}$  responds as follows:

1. If the query  $(ID_i, M_i)$  already appears on the  $H$ -list in a tuple  $(ID_i, z_i, M_i, w_i, b_i, c_i)$  then  $\mathcal{B}$  responds with  $H(ID_i) = z_i \in Z_q^*$  and  $H(M_i) = w_i \in G_1$ .
2. Otherwise,  $\mathcal{B}$  generates a random coin  $c_i \in \{0, 1\}$  so that  $\text{Pr}[c_i = 0] = 1/(q_T + 1)$ . And  $\mathcal{B}$  picks random values  $z_i \in Z_q^*$  and  $b_i \in Z_q^*$ .
  - If  $c_i = 0$ ,  $\mathcal{B}$  computes  $w_i = b_i \cdot P + Q \in G_1$ .
  - else if  $c_i = 1$ ,  $\mathcal{B}$  computes  $w_i = b_i \cdot P \in G_1$ .
3.  $\mathcal{B}$  adds the tuple  $(ID_i, z_i, M_i, w_i, b_i, c_i)$  to the  $H$ -list and respond to  $\mathcal{A}$  by setting  $H(ID_i) = z_i \in Z_q^*$  and  $H(M_i) = w_i \in G_1$ .

**T-queries:** Let  $(ID_i, M_i)$  be a token query issued by  $\mathcal{A}$ .  $\mathcal{B}$  responds to this query as follows:

1.  $\mathcal{B}$  runs the above algorithm for responding to  $H$ -queries to obtain  $z_i$  and  $w_i$ . If  $c_i = 0$  then  $\mathcal{B}$  reports failure and terminates.
2. Otherwise, we know  $c_i = 1$  and hence  $w_i = b_i \cdot P$ . Define  $\text{Token}_i = \frac{b_i}{z_i} \cdot A\text{-ID} + \frac{r b_i}{z_i} \cdot P \in G_1$ . Observe that  $\frac{a}{z_i} \cdot w_i + \frac{r}{z_i} \cdot w_i = \frac{a+r}{z_i} \cdot w_i = \frac{a+r}{z_i} \cdot H(M_i)$  and therefore  $\text{Token}_i$  is a valid token on  $ID_i$  and  $M_i$  under the public  $\text{Auth-ID}_1 = a \cdot P + r \cdot P = (a + r) \cdot P$ .  $\mathcal{B}$  gives  $\text{Token}_i$  to  $\mathcal{A}$ .

**Output:** Eventually  $\mathcal{A}$  produces a message-token pair  $M_f, Token_f$  such that no  $T$ -query was issued for  $M_f$ . If there is no tuple on the  $H$ -list containing  $M_f$  then  $\mathcal{B}$  issues a query itself for  $H(M_f)$  to ensure that such a tuple exists. We assume  $Token_f$  is a valid token on  $M_f$  under the given public information; If it is not,  $\mathcal{B}$  reports failure and terminates. Next,  $\mathcal{B}$  finds the tuple  $(ID_f, z, M_f, w_f, b_f, c_f)$  on the  $H$ -list. If  $c_f = 1$  then  $\mathcal{B}$  reports failure and terminates. Otherwise,  $c_f = 0$  and therefore  $H(M_f) = w_f = b_f \cdot P + Q$ . Hence  $Token_f = \frac{a+r}{z} \cdot (b_f \cdot P + Q) = \frac{a+r}{z} \cdot b_f \cdot P + \frac{a+r}{z} \cdot Q$ . Then  $\mathcal{B}$  outputs the required  $\frac{a}{z} \cdot Q$  as

$$\frac{a}{z} \cdot Q \leftarrow Token_f - \frac{b_f}{z} \cdot A-ID - \frac{rb_f}{z} \cdot P - \frac{r}{z} \cdot Q.$$

This completes the description of algorithm  $\mathcal{B}$ .

It remains to show that  $\mathcal{B}$  solves the given instance CDH problem with probability at least  $\epsilon'$ . The probability analysis is identical to that in Boneh et al.'s scheme[4] and therefore omitted.

**Accountability:** An authorized subject (delegator) delegates its authority to a foreign subject by creating a delegation token using the master identity of the delegator. Since the master identity is secret information only known to the delegator, the delegator cannot repudiate the fact of delegation. Then, the service provider can know it from the delegation token that the delegation is carried out between the two subjects.

**Revoking:** If delegator with  $ID_A$  wants to revoke a delegation to a certain device  $DC$  during a certain service is provided to  $DC$ , then the delegator can create a revocation token for  $DC$ . The revocation token generation protocol is identical to  $GEN_{AToken}$  except that the delegator generates a revocation-request message  $revoke-req = 'revocation-request' || ID_A || DID_{DC} || Time_i$ , and that  $RT_A = a \cdot H_1(revoke-req)$ , where  $a = \frac{H_2(MID_A)}{H_2(DID_{DC})}$ . If the delegator sends  $\langle revoke-req, RT_A \rangle$  to the  $SP$ , then  $SP$  can stop to providing the service to the corresponding device immediately after checking the token.

**Replay Attacks:** Whenever generating authentication tokens or delegation tokens, an authorized subject creates a new `access-req` or `auth-deleg-msg` that includes current time information. Therefore, the service provider can protect the replay attack weakly by checking the time information. The service provider can deny the token authentication for the tokens that are arrived at later than communication or computation delay time pre-defined by the  $SP$ .

## 5 Conclusion

In this paper, we introduced an ID-based access control model adequate for ubiquitous computing. We eliminated the use of certificates and proposed more flexible access control model by allowing subject-oriented authority delegations using the identities of subjects. Users can enjoy certain services in anywhere

and any time using any devices by generating authentication tokens or delegation tokens. Our tokens provide device authentication and user authorization simultaneously. Therefore, service providers can easily authenticate subjects just by verifying the tokens. We implemented the token generation protocols based on pairing and proved the security of our proposed model against the chosen-message attack under the random oracle model.

## References

1. Barreto, P., Kim, H., Lynn, B. and Scott, M.: Efficient Algorithms for Pairing-based Cryptosystems. *Advances in Cryptology - Crypto'02*. LNCS 2442 (2002) 354-368
2. Blaze, M., Feigenbaum, J. and Lacy, J.: Decentralized Trust Management. *Proc. of the IEEE Conference on Security and Privacy*, (1996)
3. Boneh, D. and Franklin, M.: Identity-Based Encryption from the Weil Pairing. *Advances in Cryptology - Crypto'01*. LNCS 2139 (2001) 213
4. Boneh, D. Lynn, B. and Shacham, H.: Short Signature from the Weil Pairing. *Advances in Cryptology - Asiacrypt'01*. LNCS 2248 (2001) 514-532
5. Campbell, R., Sturman, D. and Tock, T.: Mobile Computing, Security and Delegation. *Proc. of the International Workshop on Multi-Dimensional Mobile Communications*, (1994)
6. Cha, J.C. and Cheon, J.H.: An Identity-Based Signature from Gap Diffie-Hellman Groups. *Proc. of PKC03*. LNCS 2567 (2003) 18-30
7. Ellison, C. M., Frantz, B., Lampson, B., Rivest, R., Thomas, B. M. and Ylonen, T.: Simple Public Key Certificate. Internet-Draft draft-ietf-spki-certstructure- 05.txt, Internet Engineering Task Force, (1998)
8. Gasser, M. and McDermott, E.: An Architecture for Practical Delegation in a Distributed System. *Proc. of the Symposium on Security and Privacy*. (1990) 20-30
9. Goldwasser, S., Micali, S. and Rivest, R.: A Digital Signature Scheme Secure against Adaptive Chosen-Message Attacks. *SIAM J. Computing*, 17(2) (1988) 281-308
10. Gutmann, P.: PKI: Its not Dead, Just Resting. *IEEE Computer*. vol. 35, no. 8 (2002) 41-49
11. Kagal, L., Finin, T. and Joshi, A.: Moving from Security to Distributed Trust in Ubiquitous Computing Environment. *IEEE Computer* (2001)
12. Nikander, P.: An Architecture for Authorization and Delegation in Distributed Object-Oriented Agent Systems. Ph.D. Thesis, Helsinki University of Technology, (1999)
13. Shamir, A.: Identity-Based Cryptosystems and Signature Schemes. *Advances in Cryptology - Crypto84*. LNCS 0196 (1984) 47-53
14. Sollins, K. R.: Cascaded Authentication. *Proc. of the Symposium on Security and Privacy*, (1988) 156-163.
15. Steiner, J. G., Neuman, C. and Schiller, J. I.: Kerberos: An Authentication Service for Open Network Systems. *Proc. of the Winter USENIX Conference*, (1998) 191-202
16. Zimmermann, P. R.: *The Official PGP User's Guide*. MIT Press, (1995)
17. Public-Key Infrastructure (X.509), <http://www.ietf.org/html.charters/pkix-charter.html>

# How to Construct Secure Cryptographic Location-Based Services

Jun Anzai<sup>1,2</sup> and Tsutomu Matsumoto<sup>1</sup>

<sup>1</sup> Yokohama National University, 79-5, Tokiwadai, Hodogaya, Yokohama 240-8501, Japan  
{anzai, tsutomu}@mlab.jks.ynu.ac.jp

<sup>2</sup> Panasonic Mobile Communications Co., Ltd., 600, Saedo-cho,  
Tsuzuki-ku, Yokohama 224-8539, Japan  
anzai.jun@jp.panasonic.com

**Abstract.** Recently, ubiquitous computing / networks have been studied actively. These networks provide services depending on real environments of mobile nodes. Especially, we expect location-based services (LBSs), which rely on location of mobile nodes, are anticipated to come into wide use in the future. High-value SBSs require cryptography to ensure security. Here, cryptographic SBSs comprise a key management function (e.g. key sharing with nodes) and a location management function (e.g. location verification of nodes). Cooperation between key and location management functions realizes cryptographic SBSs. However, these functions have mostly been studied individually. This study indicates that cryptographic SBSs are insecure if the cooperation is incomplete, and proposes a method of constructing secure cryptographic SBSs.

## 1 Introduction

### 1.1 Background

Recently, services using the real context of mobile nodes are actively studied on ubiquitous computing. We expect services, which use location information of nodes as a real context, to come into wider use in the future. Such services are called location-based services (LBSs). SBSs include walker navigation, mobile node tracking [9] [15], location-based access control [6], along with other applications. Many high-value SBSs require security; for that reason, SBSs require cryptographic capability. As instances of cryptographic SBSs, we postulate a system in which a user can read a secret business document stored in a notebook PC in an office. In this system, re-encryption of the document prevents its reading when the PC is removed from the office. Cryptographic SBSs comprise a key management function (e.g. key sharing for session encryption) and a location management function (e.g. node location verification). Key management methods have been studied variously until now. As location management, location measurement methods using global position system (GPS) and radar are realized; methods using wireless LAN and radio frequency identification (RFID) are advancing apace. However, these management methods have mostly been studied individually. Therefore we consider security of integration between key management and location management for realizing secure cryptographic SBSs.

## 1.2 Location Management

Location measurement technologies have made the transition from methods [7] using GPS, a base station (of cellular phones) and radar for outdoors to methods [10][11][13] using wireless LAN, RFID for indoors. Papers [2][3][5][12][14] proposed secure location verification schemes using communication delay. This paper refers to digitized location information as a “Location Token”. We assume a location token model that composes plural location token providers (with various location measurement functions), provers (which prove their own location using the tokens) and verifiers (which verify prover location using the tokens). One study [14] adopts a narrowly-defined location token mode that supposes one location measurement scheme. This paper uses location management as a generic term to refer to the following functions:

- Location verification: a verifier directly verifies a mobile node location in real-time.
- Location certification: a verifier indirectly verifies a mobile node location using location tokens.

## 1.3 Key Management

In this paper, key-management targets are the following keys: **a client private key**: is a unique secret key of a mobile node (and the corresponding public key), and **a processed key**: is output of a key-management function that is an inputted client private key dependent on information, the output is a secret key (and the corresponding public key). Key management is a generic term used to refer to the following functions:

- Key issuing: is a method that issues a key, e.g., issuance of a public key certificate;
- Key sharing: is a method by which plural entities share the same key, for instance Diffie-Hellman key exchange scheme;
- Key distribution: is a method that distributes a key to specific entities, for instance broadcast encryption,
- Key generation: is a method that one or more entities generate a key, for instance RSA key generation,
- Key revocation: is a method that revokes a key, for instance broadcast exclusion and Certificate Revocation List; and
- Key control: is a method that controls access to a key, for instance Kerberos.

## 1.4 Integration of Key Management and Location Management

One study [6] proposed a PC system, in which a PC hard disk is decrypted because a personal radio device allows the PC to use a decryption key if their authentication is successful, when the device closes in the PC. Another study [4] proposed trusted access points measuring mobile node location and shares a key using electric field intensity of beacons sent by the points. The studies are schemes that mix location management and key management functions. However the studies do not clarify the structure of integration between the two functions. Thus we cannot analyze structural security. Incomplete integration might cause an attack on cryptographic LBSs, e.g., a provider would like to share a key with a mobile node on a specific location. However, an attacker on other location may force the provider to share the key with the at-

tacker after the valid node location verification. Therefore, we consider security of integration between key management and location management for realizing secure cryptographic LBSs. This paper treats three problems: 1) an attacker may impersonate a valid node if target nodes are not the same on key management and location management functions; 2) an attacker may replace a valid function with an invalid function if key management and location management functions are indivisible; and 3) a provider may not provide valid LBSs if a function execute after execution of another function is a failure.

## 1.5 Our Goal

Cryptographic LBSs require key management and location management functions. Secure integration of these functions has not clarified because the functions have only studied individually until now. This paper proposes a method of integrating key management and location management functions for realizing secure cryptographic LBSs. In addition, we suggest new cryptographic LBSs by assessing combinations of key management and location management. Our proposed method defines two general integrated functions (location key function). Our method consists of a location key server that provides services using location key functions and a location key client that requests a service to the server. For realizing secure cryptographic LBSs, we apply approaches: 1) agreement of target nodes by a context connection (CC) value; 2) improvement of mutual dependence by a construct inputs output of a management function to another management function; and 3) policy-based access control for functions.

## 2 Definitions and Requirements

### 2.1 Location Key

In this paper, a location key function means an integrated function of key management and location management functions. The location key function is classified as follows:

- **LK** (Location operation, then **K**ey operation) function - is a key management function that is inputted to output of a location management function; and
- **KL** (**K**ey operation, then Location operation) function - is a location management function that is inputted to output of a key management function.

LK and KL functions output a pair of a location key and its corresponding location token. Note that an LK function does not output a location token basically. A location key is a processed key: that is outputted from an LK function; or is targeted by a location token. The location keys are determined from location between a client location, a time when location key function is executed, and a client private key.

### 2.2 Entities

Our proposed method consists of the following entities:

- **Location key client:** is a mobile node that requests services using location key functions to a location key server. A client stores a unique client private key se-



curely and his ID is “i”. In addition, the client might obtain his own location information, time information, and random numbers. This study supposes a cellular phone, PDA and a notebook PC as clients.

- **Location key server:** is a server that provides services using location key functions to a location key client. A server has KL and LK functions as location key functions, and stores a unique server private key securely; the ID is “j”. The server provides services using the location key functions according to a location key policy to a location key client. A server might obtain its own location information, time information, and random numbers. This study supposes the following as servers: 1) **Station:** is a trusted apparatus that has high performance and is fixed on a specific location. This study presumes a base station of cellular phone systems and an access point (or a PC connecting to the point) of a wireless LAN as a station; 2) **Mobile:** is a mobile node that has middle performance. This study supposes a notebook PC, a PDA, or a cellular phone as a mobile. A location key client does not necessarily trust a mobile; 3) **Sensor:** is a fixed node that has low performance and is active. This study supposes a sensor node of sensor networks and an active IC tag (e.g. a smart tag) of RFID as a sensor. A sensor is not highly trusted by a location key client; 4) **Tag:** is a device that has little performance and is passive. This study supposes a tag of RFID as a tag. A location key client does not completely trust. In case of wearing a tag, a reader/writer writes a location token to the tag. Therefore, the tag is a location key client and the reader/writer is a location key server.

### 2.3 Location Measurement Methods

We classify methods that measure client locations into the following types: 1) **Report type:** means a method in which a client reports self-obtained location of the client to a server. This study presumes a client supporting system, for example GPS; 2) **Inference type:** means a method in which a server infers location of a client from evidence (e.g. IDs of tags). This study generally presumes methods using RFID; 3) **Direct type:** means a method in which a server directly verifies location of a client in real-time. This study presumes methods using radar, a wireless LAN, and communication delay.

### 2.4 Location Token

A location token is digitized location information obtained from a location key server with location measurement methods. The token includes a location key client ID, client location information and a time when a location key function executes. The token also includes a location key, or a location key function issues a pair of a location token and the corresponding location key. This study assumes the following location tokens.

- **A location certificate:** is a kind of public key certificate with which a station or a mobile signs IDs of a client and a server; client location information, a location key and a time when a location key function executes. Apparently, a location certificate is a kind of time-stamp [8] that includes location information and a location key. The location certificate supposes direct type location measurement methods.
- **Location evidence:** is digitized location information with a message authentication code that a location key client obtains from a sensor. Use of location evidence supposes inference-type location measurement methods. The location evidence in-

cludes a sensor ID or client location information. The location evidence might include a client ID, a location key and a time when a location key function executes.

- **Provisional location evidence:** is digitized location information that a location key client obtains from a tag. The provisional location evidence supposes inference-type location measurement methods and includes a tag ID.
- **A location reference:** is digitized location information that a location key client self-calculates using supporting entities (e.g. GPS). The location reference supposes report-type location measurement methods.

Table 1 show the relation between location tokens, servers and location measurement methods. A hyphen means that the corresponding method is nonexistent now. Define the corresponding new location token if a method that corresponds to the hyphen appears. A station, a mobile and a sensor receive a location reference and can then transform the location reference to a location certificate or evidence.

**Table 1.** Relation between location tokens, servers and location measurement methods

Method Server	Report type measurement	Inference type measurement	Direct type measurement
Station	Location reference	-	Location certificate
Mobile	Location reference	-	Location certificate
Sensor	Location reference	Location evidence	-
Tag	-	Provisional location evidence	-

### 2.5 Security Assumptions

This paper makes the following security assumptions:

1. A location key server becomes trust, a station, a mobile, a sensor and a tag in that order. Especially the station is a trusted party.
2. Each key management method and each location management method is secure.
3. An attacker is a location key client or a third party.
4. For attacking, a location key client and a third party might conspire.
5. A communication channel is not secure: anyone can obtain data on the channel.
6. An attacker purposes location key functions of a location key server to use illegally, and purposes outputs of the location key functions to use change illegally.

### 2.6 Requirement

This paper designs our proposed method for satisfying the following requirements:

1. **Availability:** is that only allowed location key clients can use location key functions of a location key server according to a location key policy.
2. **Universality:** is that our method is easily adaptable to existing systems. Actual systems, which include various servers, location measurement technologies, key and location management methods, require our method to universal design.
3. **Associativity:** is that association between a key management function and a location management function is secure. Consequently, our proposed method solves three problems shown in section 1.4.

4. **Privacy:** is that information (a client ID, client location information and a time when a location key function executes), which is demanded by a client demands to conceal, is not leaked from a location token. A server excludes inconsistent requests of a client. This paper respectively refers to concealing an ID, concealing client location information, and concealing a time when a location key function executes “anonymity, location-hiding and time-hiding” respectively.

### 3 Proposed Method

#### 3.1 Notation

We next show the notation for explaining our proposed method:

- **CK:** is a client private key. **SK:** is a server private key. **R:** is a random number.
- **CID:** is an ID of a location key client. **SID:** is an ID of a location key server.
- **Info<sub>C</sub>:** is location key client information that consisting of R, location information and time information. **Info<sub>S</sub>:** is location key server information that comprising of R, location information, and time information.
- **Data<sub>CK</sub>:** is data that depends on CK (and Info<sub>C</sub>). **PK:** is a processed key that is the output of a key management function with input is Data<sub>CK</sub>.
- **LKP:** is a location key policy that includes conditions for executing location key functions. **LKS:** is an internal status of a location key server: the status (e.g. existence of a specific key, and current location of a server) is needed for judging LKP.
- **KMT:** is a type of key management (key issuing, key generation, key sharing, key distribution, key revocation, key control and none). **LMT:** is a type of location management (location verification, location certification and none). **LKT:** is a type of location key function (KMT || LMT and LMT || KMT).
- **LTT:** is a type of location token (a location certificate, location evidence, a location reference, and none). **CPT:** is a type of client privacy (anonymity, location-hiding, time-hiding, and none).
- **CC value:** is a context connection value (CID, R, PK, or location token).
- **KM function:** is a key management function that outputs PK for input (Data<sub>CK</sub>, KMT, CID, SID, SK, Info<sub>C</sub>, and Info<sub>S</sub>). According to KMT, {CID, SID, SK, Info<sub>C</sub> and Info<sub>S</sub>} cannot be ignored. As PK, a KM function outputs CID (in case that KMT indicates a key management method with client authentication) or R (in case that Info<sub>C</sub> or Info<sub>S</sub> include R).
- **LM function:** is a location management function that outputs a location token for input (Data<sub>CK</sub>, LMT, SID, SK, Info<sub>C</sub>, Info<sub>S</sub>, and LTT). According to LMT, {CID, SID, SK, Info<sub>C</sub> and Info<sub>S</sub>} cannot be ignored. The LM function outputs a pre-selected location token if LTT is none. As a location token, the LM function outputs CID (in case that LMT indicates a location management method with client authentication) or R (in case that Info<sub>C</sub> or Info<sub>S</sub> include R).
- **PJ function:** is a policy judgment function that outputs KMT, a pair of {CPT, LMT, LTT} or Reject for input (LKP, CPT, LTT, LKS, LKT, CID, and Info<sub>C</sub>). According to LKP, {CPT, LTT, LKS, LKT, CID, and Info<sub>C</sub>} cannot be ignored.
- **KL function:** is a location key function that outputs a pair of {a location key, a location token} or Reject for input (Data<sub>CK</sub>, LKP, LKS, LKT, CID, SID, SK, Info<sub>C</sub>,

Info<sub>S</sub>, LTT, and CPT). A location token includes a location key if LTT = “location certificate”. According to LKT and LKP, {LKS, LKT, CID, SID, SK, Info<sub>C</sub>, Info<sub>S</sub>, LTT, and CPT} cannot be ignored.

- **LK function:** is a location key function that outputs a location key or Reject for input (Data<sub>CK</sub>, LKP, LKS, LKT, CID, SID, SK, Info<sub>C</sub>, and Info<sub>S</sub>) According to LKT and LKP, {LKS, LKT, CID, SID, SK, Info<sub>C</sub> and Info<sub>S</sub>} cannot be ignored.

### 3.2 Constructions of a KL Function and an LK Function

Figure 1 shows constructions of KL and LK functions that comprise KM, LM and PJ functions. The KL function consists of a management path that inputs output of a KM function into an LM function, and a control path that inputs output of a PJ function into KM and LM functions. On the management path, a KM function executes a key operation, which is requested by a key management type KMT, with a location key client for input (a client ID CID, a server private key SK, server information Info<sub>S</sub>, a server ID SID, and client information Info<sub>C</sub>); and then the KM function outputs a processed key PK. In addition, the KM function outputs CID (if the KM function performs client authentication) or a random number R (if the KM function performs client distinguishing with a temporal ID (i.e. R)) as CC values. The PK is a CC value if the KM function does not output CID or R. Next, an LM function executes a CC value-dependent location operation, which is requested by a location management type LMT, with a client for the output of the KM function, and then the LM function outputs a pair of {a location key, a location token} that is requested by a client privacy type CPT and a location token type LTT. Here, a term “CC value-dependent” means that the LM function authenticates that the client has CID, R or the corresponding secret information. On the control path, a PJ function controls execution of a KM function according to a location key policy LKP, a location key status LKS and Request of a client. Next, the PJ function controls execution of an LM function according to the output of the KM function, LKP, LKS, and the Request. If the PJ function outputs Reject, the KL function stops execution and outputs Reject. The LK function consists of a management path that inputs output of an LM function into a KM function and a control path that inputs output of a PJ function into LM and KM functions.

On the management path, an LM function executes location operation, which is requested by LMT, with a client for input (CID, SK, Info<sub>S</sub>, SID and Info<sub>C</sub>); and then the LM function outputs a pre-selected location token. The LM function outputs CID (if the LM function performs client authentication) or R (if the LM function performs client distinguishing with a temporal ID (i.e. R)) as CC values. The location token is a CC value if the LM function does not output CID or R. Next, a KM function executes a CC value-dependent key operation, which is requested by KMT, with a client for the output of the LM function, and then the KM function outputs a location key. On the control path, a PJ function controls execution of an LM function according to LKP, LKS and the Request of a client. Here, the PJ function inputs none as CPT and LTT. Next, the PJ function controls execution of a KM function according to the output of the LM function, LKP, LKS and the Request. If the PJ function outputs Reject, the LK function stops execution and outputs Reject.

Here, LKP comprises plural records. Each record includes three items: an attribute that is client identification or a client belonging, an action that is an allowed location key functions operating, and a condition that is a requirement to allow the action.

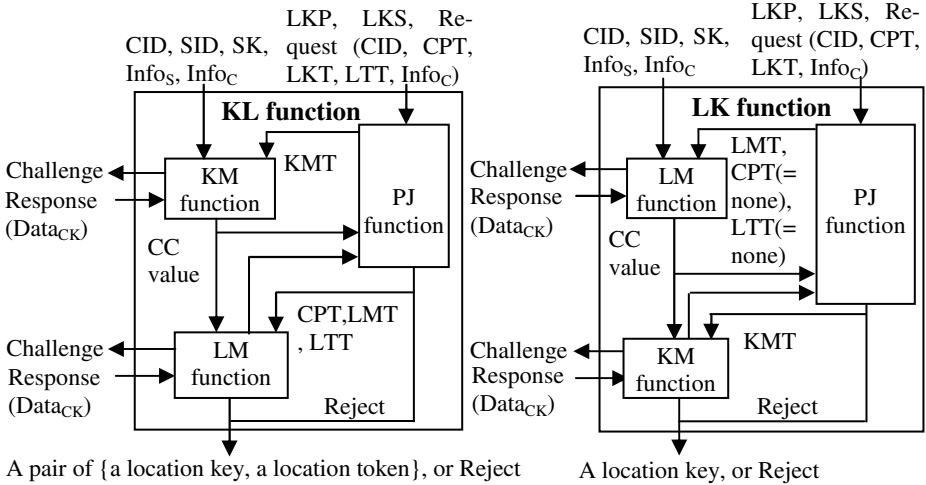


Fig. 1. Constructions of a KL function and an LK function

### 3.3 Sequence of Our Proposed Method

Figure 2 shows a sequence of our proposed method.

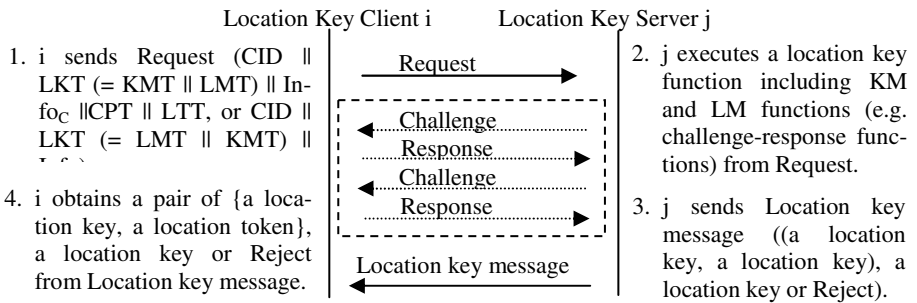


Fig. 2. Sequence of our proposed method

### 3.4 Combinations of Key Management and Location Management

We introduce instances of combinations (see Table 2) of key and location management, and believe that other concrete schemes exist in addition to the following instances.

- LK function 2: is a scheme that a server verifies client location and shares a symmetric key with the client if the location is within 50[m] (e.g. the scheme [4] exists).
- KL function 3: is a group key generation in which group members can certificate who, where, and when to share the key with a third party, using the scheme [1].
- LK function 4: is a scheme by which a server verifies the client location; then the server distributes a decryption key for encrypted data (e.g. business documents) if the location is in the specific area (e.g. an office). For example, scheme [6] exists.

**Table 2.** Combinations of key management and location management

Key \ Location	Location verification		Location certification	
Key issuing	KL function 1	LK function 1	KL function 7	LK function 7
Key sharing	KL function 2	LK function 2	KL function 8	LK function 8
Key generation	KL function 3	LK function 3	KL function 9	LK function 9
Key distribution	KL function 4	LK function 4	KL function 10	LK function 10
Key revocation	KL function 5	LK function 5	KL function 11	LK function 11
Key control	KL function 6	LK function 6	KL function 12	LK function 12

## 4 Evaluation

### 4.1 Viability of Our Proposed Method

This section shows that our proposed method satisfies the following requirements.

**Availability:** A location key server has location key functions; a PJ function can control those functions using a location key policy. Consequently, only the allowed location key client can use the location key functions. Here, a KM function or an LM function (in the location key functions) authenticates client identification.

**Universality:** As shown in Table 1, our method assumes four location token types that correspond to combinations of four location key server types and three location measurement method types. A location key client can request a provided location token type to the server. Therefore, various existing systems can adopt our method. Moreover, existing key and location management schemes can apply to a KM and an LM functions because our method treats the KM and LM functions as modules.

**Associativity:** From Figure 1, a KL function structure can force output of a KM function to be input of an LM function for boosting the relation between the KM and LM functions. In the same way, an LK function structure can force output of an LM function to be input of a KM function. On location key functions, KM and LM functions can authenticate the same client using CC values, for preventing differences between clients whom the KM and LM functions authenticate. Here, a CC value (CID, R, PK, and a location token) depends on a client by which each management function targets as follows: 1) a server can identify a client because CID is a unique ID; 2) a server can distinguish a client that has R from another client that does not have the R. But the server cannot identify a client because the R is a temporal ID by which the server gives for providing LBSs; 3) a server can distinguish a client that has PK from another client that does not have the PK. But the server may not be able to identify a client because a key management method may not authenticate the client on a KM function; 4) a server can distinguish a client that has a location token from another client that does not have the token. But the server may not be able to identify a client because the server may not identify the client for location management. Thus, a server can authenticate the same client using a CC value when the client requests anonymity on client authentication, and when management methods of LM and KM functions do not support a temporal ID. Here, securities of CC values rely on that of the methods. In addition, location key

functions inputs are limited to a Request and Responses. A PJ function can verify the Request directly. On the other hand, KM and LM functions verify the Responses; a PJ function verifies feedback from the KM and LM functions. In a word, the PJ function can verify the Responses indirectly. The PJ function can also stop execution of a location key function if the PJ function receives feedback that the KM and LM function are unable to authenticate the same client. Therefore, the PJ function can verify a management path between the KM and LM functions.

**Privacy:** A server outputs an only location token as client privacy information. A client can demand anonymity, location hiding and time hiding to the server using a CPT. The server generates the token, which is excluded privacy information selected by the CPT, if a PJ function allows the token generation from a LKP.

## 4.2 Security Analysis

We analyze security of our method. From section 2.5 and 4.3, location key functions are secure for external attacks. In addition, a location key server is a trusted party when the server is a station. Consequently, our method can prevent attacks described in section 2.5 if a location key server type is a station. The security of our method is equal to trustiness of a server when the server type is a mobile, a sensor, or a tag.

## 5 Conclusion

This paper proposed a method of constructing secure cryptographic LBSs, which have a location key function consisting mainly of a location management function and key management function. Our proposed method includes three approaches: apply a construct by which output of a management function inputs another management function, context connection value, and policy-based access control to location-key functions.

## References

- [1] J. Anzai, T. Matsumoto, "Interaction Key Generation Schemes," *IEICE Trans. Fundamentals*, Vol. E87-A, No. 1, pp. 152-159, 2004.
- [2] J. Anzai, T. Matsumoto, "Location Verification (1): Location Verification Schemes Resistant Against Relay Attack," *Proc. of SCIS2005*, 2B4-3, 2005.
- [3] S. Brands, D. Chaum, "Distance-Bounding Protocols," *Proc. of Eurocrypt'93*, Springer-Verlag, pp. 344-359, 1993.
- [4] S. Banerjee, A. Mishra, "Secure Spaces: Location-based Secure Wireless Group Communication," *Mobile Computing and Communications Review*, Vol. 1, No. 2, 2002.
- [5] S. Capkun, J. P. Hubaux, "Securing position and distance verification in wireless networks," *Technical report EPFL/IC/200443*, 2004.
- [6] M.D. Corner, B.D. Noble, "Zero-Interaction Authentication," *Proc. of MOBICOM2002*.
- [7] E. Gabber, A. Wool. "How to prove where you are: Tracking the location of customer equipment," *Proc. of 5th ACM Conf. Computer and Communications Security*, pp. 142-149, 1998.
- [8] S. Haber, W. S. Stornetta, "How to Time-Stamp a Digital Document," *Journal of Cryptology: the International Association for Cryptologic Research* 3, 2, 99-111, 1991.

- [9] M. Izumi, S. Takeuchi, Y. Watanabe, K. Uehara, H. Sunahara, J. Murai, "A Proposal on a Privacy Control Method for Geographical Location Information Systems," Proc. of INET2000.
- [10] T. Kitasuka, T. Nakanishi, and A. Fukuda, "Indoor Location Sensing Technique using Wireless Network," Proc. of Computer System Symposium'02, pp. 83-90, 2002.
- [11] K. Nakanishi, J. Nakazawa, and H. Tokuda, "LEXP: Preserving User Privacy and Certifying the Location Information," 2nd Workshop on Security in Ubicomp2003.
- [12] N. Sastry, U. Shankar, and D. Wagner, "Secure Verification of Location Claims," Report No. UCB//CDS-03-1245, University of California, Berkeley.
- [13] A.Vora, M.Nesterenko, "Secure Location Verification Using Radio Broadcast," Proc. of OPODIS2004:8th International Conference on Principles of Distributed Systems, 2004.
- [14] B. R. Waters, E. W. Felten, "Secure, Private Proofs of Location," Princeton University Computer Science Technical Reports, TR-667-03, 2003.
- [15] Y.Watanabe, S.Takeuchi, F.Teraoka, K.Uehara, and J.Murai, "The Geographical Location Information System with Privacy Protection," IPSJ Journal, Vol. 37, No. 6, 1996.



# A Key Management Scheme for Mobile Ad Hoc Networks Based on Threshold Cryptography for Providing Fast Authentication and Low Signaling Load

Hoang Nam Nguyen<sup>1</sup> and Hiroaki Morino<sup>2</sup>

<sup>1</sup> The National Institute of Information and Communication Technology,  
4-2-1 Nukui-kitamachi, Koganei-city, Tokyo 184-8795, Japan  
nguyen@nict.go.jp

<sup>2</sup> Shibaura Institute of Technology,  
3-9-14 Shibaura, Minato-ku, Tokyo 108-8548, Japan  
morino@sic.shibaura-it.ac.jp

**Abstract.** Providing secure communications is a crucial task for the success of future ubiquitous mobile communication systems. Using public key infrastructure (PKI) is considered as a good solution to fulfill the task. However, as mobile ad hoc networks (MANET) inherit unique characteristics such as dynamic topology, non-infrastructure architecture, centralized PKI architectures are not suitable for dynamic MANET. The use of distributed PKI models is more appropriate but requires additional modifications to adapt with network changes. In this paper, we introduce a novel key management scheme for MANET, which exploits advantages of threshold cryptography. The major innovative aspect of this scheme is the use of temporal substitute certificate authorities (SCA), which form a PKI model of multi SCA groups. Performance results obtained by computer simulation show that the proposed key management scheme can reduce the latency of authentication, certificate update delay and the signaling load.

## 1 Introduction

In mobile ad hoc networks (MANET), an ad hoc node operates as not only an end terminal but also an intermediate router. Data packets sent by a source node can reach to a destination node via a number of hops i.e. more than one node might be involved in forwarding packets from sources to destinations. MANET inherits unique properties such as arbitrary and dynamic network topology, user mobility and less robust wireless links. These properties bring many significant technical challenges ranging from the physical layer to the application layer for radio resource allocation, QoS control, medium access control (MAC) protocol, routing and security. Various research efforts have been carried out aiming to provide QoS guaranteed and secure communications in MANET. To provide secure communication, each MANET has to achieve security requirements in terms of availability, confidentiality, integrity, authentication and non-reputation [1]. While designing security mechanisms for MANET, following features of MANET should be taken into account: weak-secure wireless link, user roaming, dynamic topology and huge number of nodes. Efficient security schemes should be distributed to achieve high survivability. In MANET,

routing is one of the most important functions for delivering data between mobile nodes. Routing protocols in MANET are suffered various type of security threats and attacks which can come from external malicious nodes or internal compromised MANET nodes [1, 2]. In order to protect routing information, routing packets have to be coded by using a suitable public-key mechanism [3, 4, 5].

In the paper, we propose a novel key management scheme based on threshold cryptography theory for MANET. The scheme is differed to other existing schemes in terms of using temporal substitute certificate authorities (SCA). Nodes, which have more resources denoted as strong nodes, can be selected to operate as temporary CA forming multi-groups of trust servers in the network area. The use of multi SCA groups is aiming to provide short authentication delay and low signaling load. Performance evaluation obtained by computer simulation show that by using multi SCA-groups, the system is able to provide fast certificate update and low signaling load. In the next section, recent proposals for public key management schemes are presented and concepts of threshold cryptography are given. In section 3, our proposed key management model is presented where a new PKI model is introduced and authentication processes are described. Section 4 discusses the issues of a location-based SCA allocation scheme. Performance evaluation is presented in the next section. Finally, conclusion remarks are given in the last section.

## 2 Related Works

Two principles of existing key management in MANET are *node participation* and *usage of trusted third parties* [8] which are corresponding to *certificate chain* and *virtual CA* approaches. The certificate chain approach [7] requires participated nodes to have strong processing capabilities. When many nodes participate to an authentication chain, the system is more vulnerable resulting in the trade-off between the number of participated nodes and security. The usage of trusted third parties can provide the guarantee of nodes trust i.e. the authentication provided by authority entities has higher level of confidence. In MANET, the central authority might be a target for DoS and compromising attacks. When mobile ad hoc nodes locate far from the central authority, their connections might be not available.

In a threshold cryptography scheme  $(n, k)$ , a public/private key pair  $(P_k, S_k)$  of the network is generated where the  $P_k$  is distributed to all nodes in the networks [9]. The private key  $S_k$  is divided to  $n$  parts so call secret shares so that if  $k$  shares are combined, the private key is created. The nodes storing the secret shares are called as share servers or distributed certificate authorities. Public key management schemes proposed in [1] are purely based on threshold cryptography theory with enhanced properties. Authors have shown that by using distributed share servers, mobile ad hoc networks will have higher availability, higher fault tolerance and less vulnerability. A cluster-based security architecture for MANET exploiting threshold cryptography has been proposed in [6]. In this architecture, the network private key is distributed over cluster heads (CHs). When a new node enters the network, if it receives a beacon signal of a CH, it will perform a log-on procedure to become a member of the cluster. Otherwise, the node can form a new cluster and become its own cluster head. When a CH leaves or joins the network, the secret shares have to be renewed. The paper showed good performance of this architecture under a small network size.

A fully self-organized key management scheme has been proposed in [7] in which users issue certificates for each other based on their personal known information and relationship. Each user maintains a local certificate repository which stores certificates of its neighbors. When two users want to verify their public keys of each other, they try to find an appropriate chain of certificates. The scheme is suitable to open mobile ad hoc networks. However it requires particular computing power of nodes and consumes time and bandwidth. In [8] both threshold cryptography and trust chain are exploited in order to gain both high security requirements as well as the deployment for large scale MANET. The composite key management scheme introduces a security metric so called *confidence value* which defines the security level of certificate issuer. Certificates can be issued by CAs and nodes who can participate to authentication, denoted as participant nodes. CAs issue certificates to nodes those are connected directly where as participant nodes issue certificate to their neighbors. When a node tries to find a certificate chain to other node, the most appropriate chain is selected based on the concurrent confidence value of available certificate chains.

Existing key management schemes based on threshold cryptography do not consider issues of scalability, signaling load and authentication latency. In a large scale MANET, CAs can locate very far from nodes resulting in long authentication latency. When the number of nodes increases, high signaling traffic occurs. Our research objectives are to design a threshold cryptography based PKI scheme for MANET which can reduce authentication latency and signaling load.

### 3 Public-Key Management Architecture

The system model of managed open mobile ad hoc networks is shown in Fig. 1 where distributed authority entities are deployed. There are several types of ad hoc network nodes which have different node properties. Fixed and mobile stations have more capabilities of information processing and radio resource than other nodes. The stations can act as gateways connecting the ad hoc network to other external networks via wired or wireless (satellite) links. These stations can perform functionalities of routing and security. There are two classes of nodes for high power processing nodes denoted as “strong” nodes and low power processing nodes in the network.

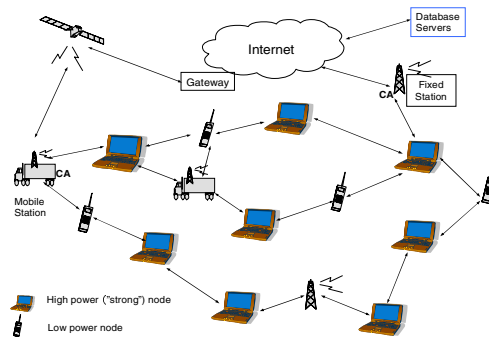
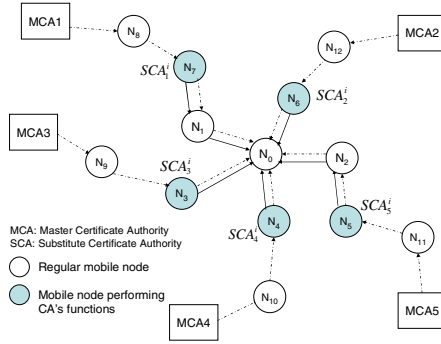


Fig. 1. System model



**Fig. 2.** Public-key management model

A novel public-key management model is shown in Fig. 2. At a given time, the network uses a pair of public and private keys ( $P_K, S_K$ ). Every node knows the public key  $P_K$  of the network whereas the private key  $S_K$  maintained by certificate authorities is used to sign certificates which prove the legal public keys of other nodes. In order to provide fast authentication and certificate update and reduce signaling load, the public key model exploits temporal substitute certificate authority (SCA) dynamically. End nodes, which have enough processing capability and resource to perform digital signature based functions, can act as SCA for a certain period. Each  $MCA_j$  has a set of  $(m + 1)$  secret shares ( $S_K^{0j}, S_K^{1j} \dots S_K^{mj}$ ) where  $S_K^{0j}$  is used by the  $MCA_j$ . The other  $S_K^{ij}$  can be distributed to corresponding  $SCA_j^i$ . That means there can be  $m$  different groups of SCA simultaneously where with the combination of  $k$  secret shares  $S_K^{ij}$  among a group, the private key  $S_K$  is created.

Assume the nodes  $N_3$  to  $N_7$  are selected by MCAs to work temporarily as SCA for a certain period. When the node  $N_0$  enters the network, it can broadcast an authentication request and receive the first reply from node  $N_3$ , and/or other nodes to inform about the list of available SCA. The  $P_{K,N_0}$  is sent to all  $SCA_j^i$  and waits for encrypted information signed by secret shares stored in the  $SCA_j^i$  denoted as  $(N_0, P_{K,N_0}, t_p, S_K^{ij})$ . When the new node  $N_0$  has  $k=3$  encrypted information's components, it can create a certificate for the node signed by the private key  $(N_0, P_{K,N_0}, t_p, S_K)$ . The authentication delay is significantly reduced comparing with that of the conventional scheme, where nodes have to connect with only MCAs for authentication. MCAs have to compute secret key's fragments, update secret shares to all  $SCA_i$  and cooperate each other to setup/release SCAs. MCAs/SCAs periodically broadcast their identity and certificates ( $Cert_{MCA}/Cert_{SCA}$ ) which prove that the nodes have legal rights to perform authentication process. Regular nodes store information of certificate authorities who can maintain node's certificate update. They have to perform the function for combining certificate share in order to create certificate. Because nodes have different roles, following types of certificates are needed:  $Cert_{MCA} = \{(MCA_{ID}, P_{MCA}, \text{"MCA node"}, S_K)\}$ ,  $Cert_{SCA} = \{(SCA_{ID}, P_{SCA}, \text{"SCA node"}, time_{exp}, S_K)\}$  and  $Cert_{node} = \{(node_{ID}, P_{node}, \text{"trust level"}, time_{exp}, S_K)\}$

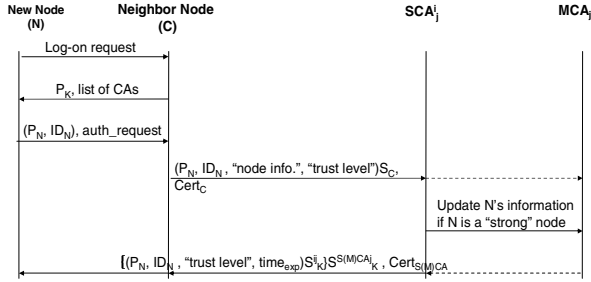


Fig. 3. Log-on authentication process

As shown in Fig. 3, when a new node N enters the mobile ad hoc network, it selects the neighbor who has the strongest signal strength and exchange information of network public key and the list of CAs. Then the node  $N_0$  sends an authentication request to this node. The neighbor node can evaluate the trust level of the new node according to the relationship between these nodes. The neighbor node adds the information of new node's trust level to this request. There are different scenarios where authentication is performed either by MCAs or by SCAs. When CAs receive this request, they will create certificate shares, sign by their private keys ( $S^{CA_j}$ ) and then send them to the new node. When the new node collects enough certificate shares, it can combine them and create its certificate. A certificate authority sends an appropriate certificate share together with the CA's certificate in order to avoid malicious nodes behaving as fake CAs. The new node can verify the CA's certificate and get the public key of the CA to decrypt the certificate share.

During its lifetime, the node N needs to update its certificate according to the validity time set by certificate authority. When a node moves within the network, its location might change frequently. The node needs to update the list of the nearest SCAs in order to obtain fast certificate update. The nodes exchange information of SCAs with each other in order to find the most appropriate SCAs for updating certificate. Each node can store the information of several SCA groups. When a node needs to update its certificate, the node can select a CAs group to ask for updating certificate. In order to reduce the signaling load, nodes can exchange information of SCA every long enough period.

### 4 Location-Based SCA Allocation

A possible and efficient SCA allocation approach is to allocate strong nodes as SCAs based on their location information. At the moment of time, mobile devices are easily equipped a GPS technology. Users can update their location information to gateway stations who also can act as MCAs. By periodically updating node's location information, MCAs can roughly draw a map of user distribution. Based on the database, MCAs can select strong nodes to work as SCAs efficiently. When a MANET is deployed to a remote area, the coverage area's geographical information of the MANET can be roughly identified and stored in MCAs. The MCAs can divide the coverage area into a number of cells for the purpose of key management. These

cells can have different sizes and are assigned certain index. The geographical map and information of cells are stored in all MCAs and can be loaded into SCAs.

When a new node enters the network, the node performs authentication and update information of its location and node properties to MCAs. The node will receive geographical cell information from MCAs. Hence when nodes move within the network, they know which cell they locate. If MCAs consider this node as a candidate node for allocating as a SCA, this node’s information (ID, location, resource) is stored in MCAs. Each  $MCA_i$  selects a “strong” node in each cell. Strong nodes having more resources (bandwidth, CPU power) are given higher priority to become a SCA. In order to avoid the case that a strong node is selected by two MCAs for a cell, MCAs exchange each other the list of nodes who are acting as SCAs (node\_ID, SCA\_group\_ID, SCA\_ID).  $MCA_i$  sends SCA\_allocation\_request to a selected node and wait for the reply. If this node is available (not yet being acts as a SCA), SCA\_allocation\_reply is sent to the  $MCA_i$ . Then  $Cert_{SCA}$  and a secret share are created and then sent to the node. When performing SCA’s secret share update and maintenance, MCAs are responsible to cooperate with each other to update certificate of SCA ( $Cert_{SCA}$ ). MCAs calculate new secret shares and deliver them to SCAs accordingly. The expiration time is the same for all SCAs belonging to a SCA\_group. Certificates of the SCAs belonging to different SCA\_group can have different expiration times.

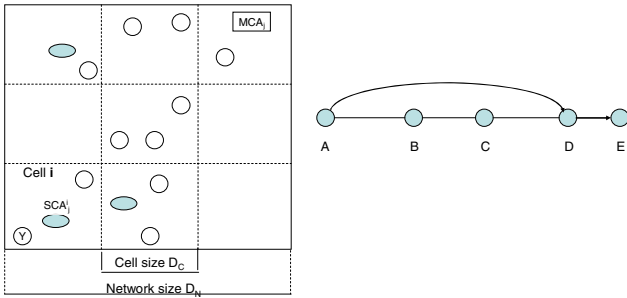
SCA handover and release process is performed periodically i.e. after every certain period of time, the MCAs will update the secret shares for the SCAs in cell  $i$  and perform SCA handover at the same time. The strong node, who is acting as a SCA of a cell but locates in another cell, will be revoked the SCA rights. The corresponding MCA of the SCA will select another strong node located in the cell  $i$  to act as a new SCA for the cell. Secret share calculation is not the major aspect of this paper therefore the detailed issues of threshold cryptography are out of our scope. Mathematic issues of secret share calculation and construction without the need of trusted dealers have been well described in [9]. In our proposed scheme, MCAs are going to calculate and update secret shares for themselves and for SCAs periodically. The share refreshing process performed for a group  $i$  is described briefly as follows:

- A set of  $k$  MCAs will involve in the share refreshing process. Each  $MCA_i$  generate a set of key shares  $(S_i^{x1}, S_i^{x2} \dots S_i^{xn})$  and delivers a  $S_i^{xj}$  to the corresponding  $MCA_j$ .

- Each  $MCA_j$  calculate a share  $S_k^{ij}$  by the following equation: 
$$S_k^{ij} = \sum_{x=1}^k S_i^{xj}$$

## 5 Evaluate the Efficiency of This PKI Model

Assume that the coverage of simulated ad hoc networks is a square of network size  $D_N$  where users are uniformly generated. MCAs divide the network coverage to  $m$  homogeneous square cells which have the cell size of  $D_c$ . The MCAs will allocate a group of SCA for each cell i.e. the  $MCA_j$  will allocate a  $SCA_i^j$  in the cell  $i$ , as shown in Fig. 4. Assume that node A wants to send packets to node E which is out of the coverage of node A. Node A can connect directly with node B, C and D. We assume that signaling packets are delivered from A to E with a minimal number of hops i.e. A->D->E. With this assumption, the delay of signaling packets transmitting between MCA/SCA and nodes depends on the distance between them. If we assume that nodes

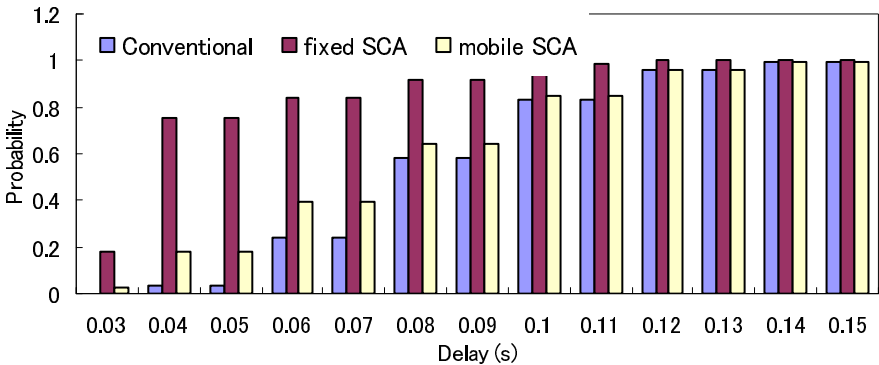


**Fig. 4.** Geographical map and packet routing

have the same maximum transmission distance ( $d_{max}$ ), a signaling packet delivered from a CA to nodes (Y), those are far from each other with the distance of  $d_{CA,Y}$ , can be transmitted roughly via  $(d_{CA,Y}/d_{max} + 1)$  hops.

Fixed MCAs are generated and their positions are uniformly distributed. After that, new nodes are generated uniformly within the network coverage. In the proposed PKI model, there are  $N_{s-node}$  strong nodes among generated nodes. The MCAs will scan each cell to find if there are more than  $n$  strong nodes in the cell and then will select  $n$  of them as SCAs. Mobile nodes move within the network with a particular mobility model (e.g. random way point). When strong nodes move to a new cell, they will update their location information to all MCAs. In our proposed PKI model, when a SCA of cell  $i$  moves to another cell, it updates its new location information to all MCA. SCA handover is not performed immediately i.e. the strong node is still operating as a SCA for the previous cell for a given time.

A network area is a square of 3000mx3000m is simulated where the maximum transmission distance of a node is assumed 300m. In most simulation scenarios if not specifically mentioned, the proposed schemes (fixed or mobile SCA) divide the networks into 16 square cells and have 100 strong nodes and exploit threshold cryptography of (5, 3). Users mobility model is random waypoint model without



**Fig. 5.** Delay's pdf of certificate update of different schemes for 1000 nodes, (5, 3) threshold

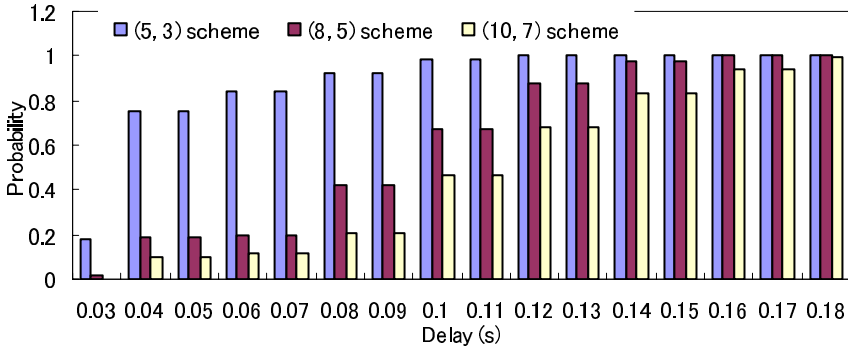


Fig. 6. Delay's pdf of certificate update: fixed-SCA scheme, 1000 nodes, different threshold

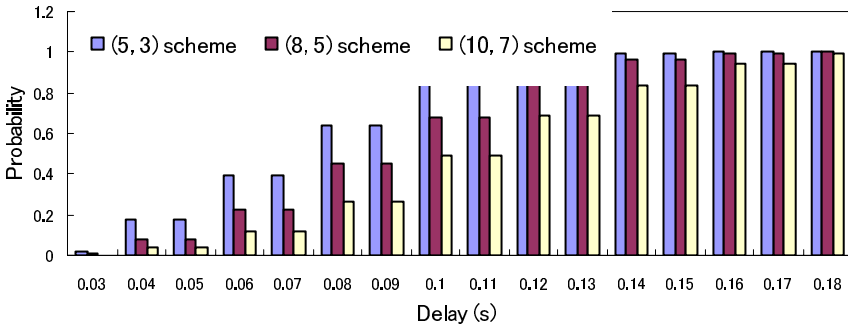


Fig. 7. Delay's pdf of certificate update of mobile-SCA scheme for 1000 nodes

pause time where user speed is varied between 1m/s to 10m/s. Without loss of generality, we select the certificate update period is 5 minutes whereas share update is 15 minutes. Delay of a transmission hop is assumed 10ms. We compare the performance of three PKI schemes: conventional threshold cryptography, proposed scheme with fixed strong nodes (fixed SCA scheme) and the proposed scheme with mobile strong nodes (mobile SCA scheme).

Fig. 5 shows that the fixed SCA scheme is the best PKI scheme in terms of certificate update delay. It can provide nearly 20% of certificate update trials with delay less than 0.02s (equivalent to 2 hops). It can provide more than 90% of updates with delay less than 0.09s. The mobile SCA scheme outperforms the conventional scheme when delay is lower than 0.09s. In the mobile SCA scheme, many trials get longer delay more than 0.09s. That is because when the strong nodes move, there are some time a cell has not enough SCAs. The nodes of this cell have to get certificate update from MCAs resulting in longer delay. Generally, by applying SCAs, the system will significantly reduce authentication latency. In Fig. 6 and 7, the certificate update delay for fixed SCA and mobile SCA schemes are presented, respectively for different threshold CA configuration. With more CA servers, system security is



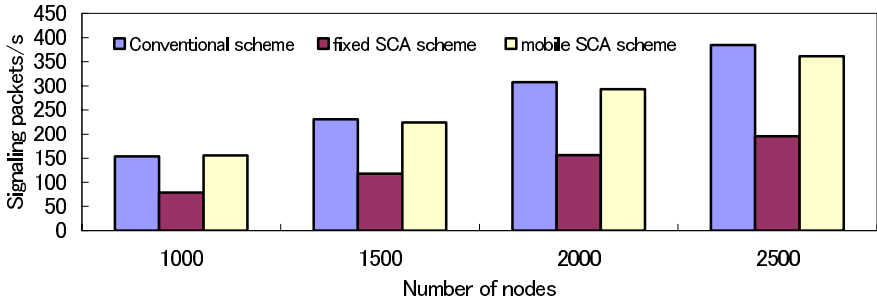


Fig. 8. Signaling load of different schemes with 16 cells, threshold (5, 3)

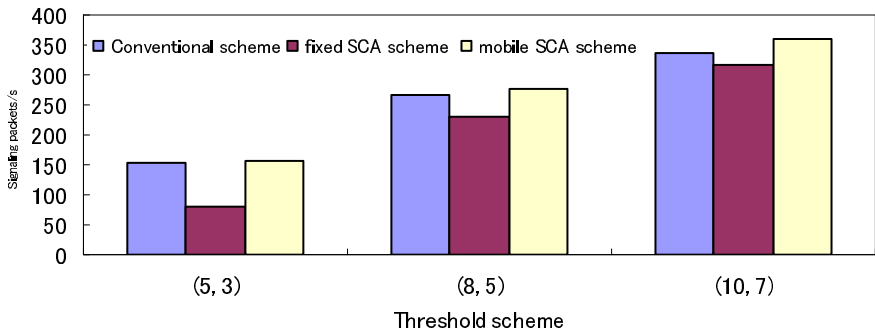


Fig. 9. Signaling load of different schemes: different threshold with 1000 nodes and 16 cells

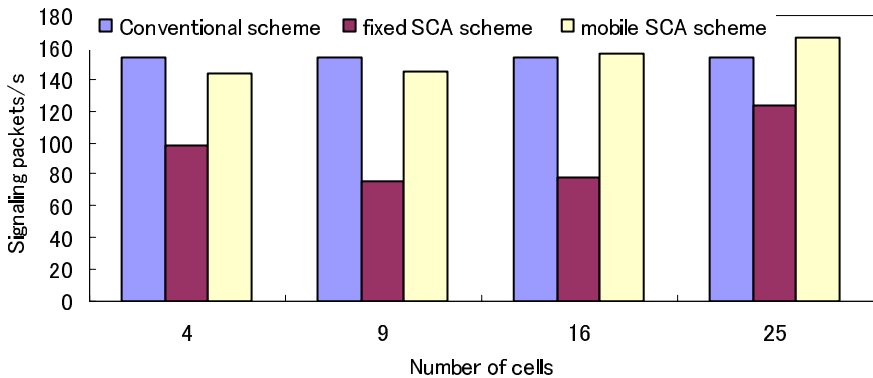


Fig. 10. Signaling load of different schemes for different cells: 1000 nodes, threshold (5, 3)

increased. When the number of CA servers increases, the performance of proposed schemes will be decreased. That is because with a fixed number strong nodes and cells, when there are more CA servers, the scheme cannot provide SCAs to certain

cells. That causes higher delay of certificate updates. Fig. 6 shows that the certificate update delay of the fixed SCA scheme decreases faster than that of the mobile SCA scheme. The reason is in the fixed SCA scheme, fixed strong nodes are allocated uniformly. With 100 strong nodes and 16 cells, there are average 6 strong nodes per cell. That means with higher threshold scheme of (8, 5) and (10, 7), many cells cannot get SCAs resulting in many long delay certificate updates. In mobile SCA scheme, when strong nodes move within the network coverage area, more strong nodes can locate in a cell. Therefore it causes higher probability that SCAs can be provided in the cell resulting in slow degradation of delay performance.

Fig. 8 to Fig. 10 show the signaling load of key management schemes for different simulation scenarios. In Fig. 8, threshold (5, 3) and 16 cells are simulated with different number of nodes. The fixed SCA schemes always provide low signaling load because fixed strong nodes do not need to update their locations and SCA handover is not necessary. When there are about 1000 nodes, the mobile SCA scheme needs signaling load similarly as that of conventional scheme. The reason is that the mobile SCA scheme has to update location of strong nodes and SCA handovers. Although it can provide low signaling load for certificate update, total signaling load of mobile SCA scheme is nearly equal to that of the conventional scheme. In Fig. 9, when different threshold configuration is applied, the fixed SCA scheme still will provide less signaling load than other schemes. However, as described above, more CA servers reduce the probability of successful SCA allocation in cells. Thus the signaling load of the fixed SCA scheme will increase fast. The mobile SCA scheme even provides more signaling load than the conventional scheme because it needs more signaling to update location of strong nodes when they move from a cell to another and more signaling load for share updates and SCA handover. Fig. 10 shows the signaling load of simulation scenarios with different cells. With the threshold (5, 3) and 1000 nodes, when the SCA-based schemes divide the network area to more cells, the signaling load of the mobile SCA scheme increases. When there are so many cells, its signaling load is even higher than that of the conventional scheme because it needs to update strong node location more frequently. The fixed SCA scheme provides higher signaling load in 4-cell scenario because, the distance between SCAs and nodes is longer resulting in more signaling for certificate updates. In the 25-cell scenario, not all cells can be provided SCAs thus resulting in more signaling load for certificate updates.

## 6 Conclusions

Using public key is an efficient solution to provide secure communication in mobile ad hoc networks. Due to the dynamic and large scale properties of MANET, providing efficient key management is a crucial task. In this paper, we have presented our public key management schemes based on threshold cryptography. Comparing with other variants of threshold cryptography, the novel proposed key management scheme exploits strong nodes as temporary substitute certificate authorities (SCA) in order to provide fast certificate update and low signaling load. Performance results show that generally the SCA-base key management schemes can gain more benefits than the conventional scheme in terms of delay and signaling load. However, there are still open research issues in terms of optimizing the SCA allocation and reducing the signaling load as well as the certificate update delay. Our future works are to optimize

the performance of the proposed schemes and evaluate their performance under different application scenarios.

## References

- [1] L. Zhou, Z. J. Haas, "Securing ad hoc networks", *IEEE Network*, Vol. 13, No. 6, Nov. 1999, pp: 24 – 30
- [2] Y. C. Hu, A. Perrig, "A survey of secure wireless ad hoc routing", *IEEE Security & Privacy Magazine*, Vol. 2, No. 3, May-June 2004, pp: 28 - 39
- [3] Y. C. Hu, A. Perrig, D. B. Johnson, "Ariadne: A Secure On-demand Routing Protocol for Ad Hoc Networks", in the Proc. Of 8<sup>th</sup> Annual International Conference Mobile Computing and Networking (Mobicom 2002), ACM Press, 20002, pp. 12-23
- [4] M. G. Zapata, N. Asokan, "Securing ad hoc routing protocols", in the Proc. Of ACM Workshop on wireless security (WiSe), ACM Press, 2002, pp: 1-10
- [5] K. Sanzgiri et al., "A secure routing protocol for ad hoc networks", 10th IEEE International Conference on Network Protocols, 2002, 12-15 Nov. 2002, pp: 78 - 87
- [6] M. Bechler, H-J. Hof, D. Kraft, F. Pahlke, L. Wolf, "A cluster-based security architecture for ad hoc networks", Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2004, Vol. 4, 7-11 March 2004, pp: 2393 – 2403.
- [7] S. Capkun, L. Buttyan, J. P. Hubaux, "Self-organized public-key management for mobile ad hoc networks", *IEEE Transactions on Mobile Computing*, Vol. 2, No. 1, Jan. 2003, pp: 52 – 64
- [8] S. Yi, R. Kravets, "Composite Key Management for Ad hoc networks", The First International Conference on Mobile and Ubiquitous Systems: Networking and Services, MOBIQUITOUS 2004, 22-26 Aug. 2004, pp: 52 – 61.
- [9] Y. Desmedt, "Some recent research aspects of threshold cryptography", In: R. Okamoto eds. *Information Security*, LNCS 1396, Springer-Verlag, 1997, pp: 158-173.

# Program Obfuscation Scheme Using Random Numbers to Complicate Control Flow

Tatsuya Toyofuku<sup>1</sup>, Toshihiro Tabata<sup>2</sup>, and Kouichi Sakurai<sup>3</sup>

<sup>1</sup> Graduate School of Information Science and Electrical Engineering,  
Kyushu University, 6-10-1 Hakozaki, Higashi-ku, Fukuoka, Japan 812-8581  
toyofuku@itslab.csce.kyushu-u.ac.jp

<sup>2</sup> Graduate School of Natural Science and Technology,  
Okayama University, 3-1-1 Tsushima-naka, Okayama, Japan 700-8530  
tabata@it.okayama-u.ac.jp

<sup>3</sup> Faculty of Information Science and Electrical Engineering,  
Kyushu University, 6-10-1 Hakozaki, Higashi-ku, Fukuoka, Japan 812-8581  
sakurai@csce.kyushu-u.ac.jp

**Abstract.** For the security technology that has been achieved with software in the computer system and the protection of the intellectual property right of software, software protection technology is necessary. One of those techniques is called obfuscation, which converts program to make analysis difficult while preserving its function. In this paper, we examine the applicability of our program obfuscation scheme to complicate control flow and study the tolerance against program analysis.

## 1 Introduction

Recently, Java, the object oriented programming language has been rapidly widespread. Java is executable in different hardware, OS, and furthermore small information terminals such as cellular phones and PDA. Described ahead, Java has a big feature of portability that it is executable on many platforms.

Java has a serious problem, however. Java program is distributed in the style called class file which is executed on a virtual machine. There is a technique called decompile that converts binary code into source code. As for the Java class file, we can easily get program code which is close to original source code. Analyzing decompiled source code, an attacker can steal algorithm used in the program code. Java has another big feature. Class file created in a certain program can be reused in the part of another program. Abusing this feature, the attacker is able to steal class file, make new program using that file, and insist on the property right of the program.

To solve these problems, software protection technique is necessary. One of those techniques is called software obfuscation. Obfuscation is a technique that converts program into another program which is difficult to analyze while preserving its function.

In this paper, we propose obfuscation scheme using random numbers to complicate control flow. We introduce how to obfuscate program control flow and study the tolerance against program analysis.

## 2 Related Works

Many obfuscation schemes have been proposed. The easiest scheme that an automatic application to the program (we call this auto-application) is called name conversion. This is a scheme of concealing what value each variable maintain and what kind of operation each function does by changing variable identifiers and function names into a quite meaningless character string. Monden et al. proposed scheme obfuscating program includes loop [1]. Ogiso et al. proved that pointers address decision problem is NP-Hard, and proposed scheme to complicate function calling by using function pointer. This scheme has theoretical proof of safety against program analysis [2]. These schemes are for obfuscating C program.

For Java program, Fukushima et al. introduced scheme to make analysis difficult by destroying the encapsulation by distributing methods. This is the scheme to destroy encapsulation which is one of the features of object oriented program and we can erase class information by this scheme [3]. This scheme is applicable to any object oriented program. Another scheme is to conceal relation between variables by linear transformation [4]. Some schemes obfuscate program by complicating control flow. For example, paper [5] proposed scheme to make analysis hard by inserting if-sentence which always returns true (or false).

To apply obfuscation scheme into huge amount of program, auto-application is required, but some scheme is difficult to do this. It is necessary to find the part where we can change program execution order while preserving original program's functionality to apply scheme introduced in paper [1]. But automation of this judging process is difficult. About scheme in paper [2], application itself is difficult because some programming language does not have a pointer.

In this paper, we propose obfuscation scheme complicating control flow. Our scheme is applicable to program written in object oriented language (we call this object oriented program), and auto-application is possible. We propose obfuscation scheme by complicating control flow and we study about execution efficiency and tolerance to the attack.

## 3 Proposed Scheme

### 3.1 Complicating Control Flow by Random Numbers

We explain our scheme using Java program. The purpose of this scheme is to complicate control flow in `main` function which exists in object oriented program, and make analysis of program's entire execution difficult.

We consider obfuscating program in figure 1. We can know an execution order of each method by analyzing `main` function part. Our scheme complicates this part and make program's entire execution flow analysis difficult. In the process of complicating control flow, we use random numbers which are difficult to predict by static analysis which is a technique to analyze program by only seeing source code. We explain an algorithm to achieve this in the following.

```

public static void main(){
(A) (B)
    method1(); method2();
    method3(); method4();}
static void method1(){
(C)
definition of method1}
//define the other methods
    
```

Fig. 1. Basic program

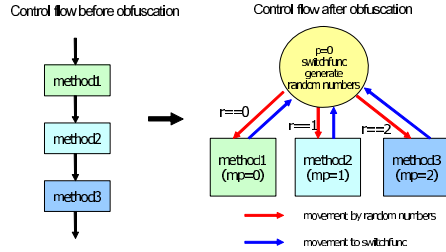


Fig. 2. Outline of method point algorithm

### 3.2 Method Point Algorithm

This algorithm consists of three steps. We introduce a detail of each step. Figure 2 shows an outline of this algorithm.

1. Setting point variable and method points
2. Generating a random number
3. Comparison of  $p$  and  $mp$

#### 1. Setting point variable and method points

Add point variable  $p$  (initial value is 0) in (A), and give a method point  $mp$  to each method.  $mp_1$  stands for  $mp$  of method1.

##### Example

$mp_1=0, mp_2=1, mp_3=2, mp_4=3$

#### 2. Generating a random number

Generate a random number  $r$  in (B). Decide to which method to move by  $r$ .

##### Example

Move to method1() if  $r$  is 0, method2() if  $r$  is 1, method3() if  $r$  is 2, and method4() if  $r$  is 3.

#### 3. Comparison of $p$ and $mp$

In the method moved in step 2, compare  $p$  and  $mp$  at (C). If both values match, execute that method, increase  $p$ , and do step 2 again. If those do not match, return to step2 without executing that method.

Repeat step 2 and 3 until  $p$  becomes 4 (the number of methods).

#### Explanation of sample behavior

There are some patterns of program behavior. We will explain them respectively.

##### Case 1: When $p$ matches $mp$ and $p$ does not become 4 after executing method

(Ex. When  $p=0$  and  $r=0$ )

Since  $mp=0$  and  $p=mp$ , execute method1 and  $p$  becomes 1. Then, regenerate random number  $r$  since  $p$  is not 4.

```

public static void main(String[] args){
    int p = 0;//initialize point variable
    switchfunc(p);}
static void method1(add variable p){
    if(p==0){//if p matches mp, execute method
    p++;//increase p}
    switchfunc(p);}
//add variable p to the other methods similarly
static void switchfunc(int p){
    if (p<4){//generate random number if program is not finished
    int r =(int)(Math.random()*4);
    switch(r){ //move to method allocated by r
    case 0:      method1(); break;   case 1:      method2(); break;
    case 2:      method3(); break;   default:     method4(); break;}}}}

```

**Fig. 3.** Outline of obfuscated program

**Case 2: When  $p$  does not match  $mp$**  (Ex. When  $p=1$  and  $r=2$ )

Since  $mp=2$  and  $p \neq mp$ , `method3` will not be executed. Regenerate random number  $r$ .

**Case 3: When  $p$  matches  $mp$  and  $p$  becomes 4 after executing method** (Ex. When  $p=3$  and  $r=3$ )

Since  $mp=3$  and  $p=mp$ , execute `method4` and  $p$  will become 4. End program because  $p$  is now 4 and this means every method is executed.

By this algorithm, we can obtain obfuscated program which has original program's function and complicate control flow. Figure 3 is an outline of program applying our scheme to program in figure1.

## 4 Expanding Proposed Scheme

We examined obfuscating program control flow by method point algorithms in the case that a program does not have such a complicate structure as branch or loop in `main` function. We call this structure simple control flow. In this section, we expand our scheme to make application possible to such a complicate control flow explained above. We consider applying our scheme to control flow in figure 4 and 5.

### 4.1 Obfuscating Branch Program

Consider the case of program control flow in figure 4. Program in figure 6 shows the sample program which has control flow in figure 4. To apply our scheme, we convert complicate program enclosed with frame into simple control flow. We can achieve this conversion of program which have branch by executing steps below.

1. Embedding branch condition
2. Allocating method point
3. Setting `switchfunc`

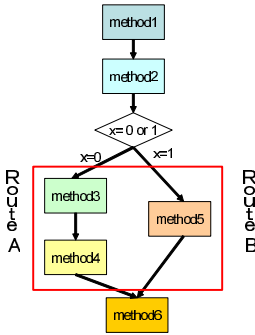


Fig. 4. Example of program control flow having branch

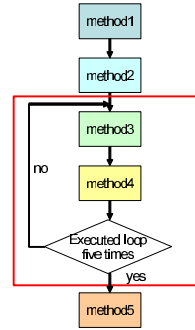


Fig. 5. Example of program control flow having loop

### 1. Embedding branch condition

Before framed part in figure 4 exists branch condition to decide whether go to route A or B. Embed this condition into method located in just before the condition (in this case, embed into method2). We also embed executing condition of each route into framed methods.

#### Example of Embedding Condition

```

int x = (int)(Math.random()*2);(I)
if (x==0){route A} else {route B}(II)
  
```

Suppose branch condition is the one written above. This means if random number  $x$  is 0, then execute route A, if  $x$  is 1, then route B. Embed condition (I) into method2 and (II) into method3, method4 (methods executed in route A) and method5 (method executed in route B).

### 2. Allocating method point

Allocate method point in executing order just like in the scheme explained in section 3.2 until reach branch point.

From the part where the branch starts to where branched execution routes join, method points are allocated from the continuation of the mp allocated in the method just before the branch point respectively.

In method executed after joining, choose the biggest mp among points allocated in the method just before joining, and allocate from the continuation of that value. Thus, each method’s mp becomes the value written below.

#### Example of allocating mp

```

mp1=0, mp2=1, mp3=2, mp4=3, mp5=2, mp6=4
  
```

### 3. Setting switchfunc

Finally, decide which method to execute by  $r$  and introduce  $p$ .

We have one point to consider. The value in  $p$  after executing each route differs. In this case,  $p$  after executing route A is 4, while after route B is 3. Method



```

public static void main(){
    method1(); method2();
int x=(int)(Math.random()*2);
    //x=0 or 1
if(x==0){
    method3(); method4();}
else {method5();}
    method6(); }
static void method1(){
    definition of method1}
//define the other methods
    
```

**Fig. 6.** Sample program having branch

```

static void method2(int p, int x){
    if(p==1){execution of method2
        p++;} //increase p
    (*)x=(int)(Math.random()*2);
    switchfunc(p,x);}
static void method3(int p, int x){
    (*)if(x==0){
        if(p==2){ //compare p and mp
            execution of method3
            p++;} //increase p
        switchfunc(p,x);}
static void switchfunc(int p, int x){
    define switchfunc as usual
}
    
```

**Fig. 7.** Outline of obfuscated program having branch

point *mp* in method executed next is 4, but after executing route B, method whose *mp* is 3 will be executed. To avoid this case, when route B is executed, adjust value added to *p* to become the next method's *mp* in the last method. If route B is executed in figure 4, *p*=2 before `method5`, so add 2 after executing method. As we see in this example, if multiple routes meet after branched, we need to adjust a value added to *p* in the method whose *mp* is smaller than other one to make *p* next method's *mp* in method just before joining of each route.

By these steps, we can apply our scheme to program which has branch structure. Figure 7 shows an outline of obfuscated program. Operation (\*) is an embedded program.

### 4.2 Obfuscating Loop Program

In this section, we consider applying our scheme to a program such have a loop repetition structure like in the part enclosed with the frame in figure 5. In this control flow, repeat `method3()`, `method4()` 5 times after executing `method1()` and `method2()`. Then, execute `method5()`. Figure 8 shows the sample program.

We can apply our scheme to control flow like figure 5 by following steps.

1. Allocating method point
2. Embedding loop finishing condition

#### 1. Allocating method point

First, allocate method point to each method as usual scheme without considering loop.

#### Example

*mp*1=0, *mp*2=1, *mp*3=2, *mp*4=3, *mp*5=4

```
public static void main(){
    method1(); method2();
for(int loop=0; loop<5; loop++){
    method3(); method4();}
    method5();}}
static void method1(){
    definition of method1}
//define the other methods
```

Fig. 8. Sample program having loop

```
static void method4(int p, int loop){
    if(p==3){
        (**)if(loop<4){
            (**)p--;
            (**)loop++;}
        else{p=p+1;}//end loop}
        switchfunc(p,loop);}
static void switchfunc(int p, int loop){
    if(p<5){int r =(int)(Math.random()*5);
        define switchfunc as usual }}
```

Fig. 9. Outline of obfuscated program having loop

## 2. Embedding loop finishing condition

Embed loop finishing condition into method executed in the last of the loop (`method4` in the case of figure 5). Explain this by loop written in `for` sentence. The `for` sentence is written in a style of (initial state; finishing condition; continuance processing). Thus, in figure 5, loop condition is `for(int loop=0; loop<5; loop++)`. This means loop will be repeated 5 times. In `method4`, compare point variable `p` and method point `mp`. If both values match, to judge whether loop is over or not after executing method, embed finishing condition and continuance processing as follows.

```
if(loop<4/*finishing condition*/) {p--;
loop++;/*continuance processing*/} else {p++;}
```

At `if` sentence, whether to continue loop or not will be judged. If continuing loop is necessary, `method3` must be executed again. Executing `method3` is impossible, however, in the time when `method4` is executed. Because at that moment, `p` is 3 and it does not match `method3`'s method point `mp3(=2)`. Therefore, we introduce new operation for `p`. If a loop must be repeated, decrease `p`. A value to decrease is equal to the number of methods in loop structure before method where the finishing condition is embedded. In this case, there are 2 methods repeated and only 1 method before `method4` where finishing condition is embedded, so we subtract 1 from `p`. Thus, embedded condition is written before. Figure 9 is an outline of obfuscated program. Operation `(**)` is an embedded program.

## 5 Evaluating Proposed Scheme

### 5.1 Attacking Program

Attack on program is divided mainly into 2 types: static analysis which analyze program only by seeing source code, and dynamic analysis by executing program. First, we examine the tolerance against static analysis. Our scheme has a feature

```
a=6, b=3
c=9
d=3
e=18
f=2
```

**Fig. 10.** Output result before dynamic analysis

```
static void sub(){
//method operates subtraction
  if(p==2){
(***)System.out.println("sub");
    execution of sub
    p++;
  }
  switchfunc(p);
}
```

**Fig. 11.** Sample of dynamic analysis

that generates random number and decides method executing next. To analyze an obfuscated program, an attacker must judge whether the called method is executed or not. He tries to find out method executing order by analyzing `p` and `mp`. Suppose analyzing `p` and `mp` is hard. In this case he takes a strategy of arranging methods suitably and analyzing the execution order. If there are  $N$  methods in the program, there are  $N!$  probable execution order. Thus, the more the method numbers are, the more the probable control flow increases, cost to static analysis grows extremely high.

Next, we consider the dynamic analysis using program computes the four basic operations of arithmetic. Figure 10 is a result of execution. For example, in figure 11, insert a program `(***)` which outputs method name when `p` matches `mp` and that method is executed. In this case, the string "sub" is displayed when executing method `sub`. Insert this program in every method changing output name. By this attack, method's name is displayed like figure 12 when each method is executed, and an attacker can know method execution order.

A method in figure 13 is considered as a countermeasure against dynamic analysis. When applying scheme, insert dummy method which has no influence on program execution result. There is no limitation in the number of dummy methods, and it can be executed many times. Example of dummy method is given in figure 13. Dummy method in figure 13 executes the following. If the condition is true, call `method4`, and if it is false, operate complicate operation for `p`. But this condition always returns false and no method is called. And complicate operation actually returns `p` itself. Moreover, the frequency of dummy method calling changes every time, analyzing method execution order using strategy considered in this section becomes difficult.

## 5.2 Program Execution Time

We applied our scheme to program computes the four basic operations of arithmetic and measured execution time. `P0'` is the program which has branch and executes route A (executes `method3` and `method4`) in figure 4, and `P0''` is the program which has loop and repeats framed part 5 times in figure 5. `P1` and `P2`, `P1'` and `P2'`, `P1''` and `P2''`, are programs obfuscated `P0`, `P0'`, `P0''` respectively. The frequency of random number generation differs. Random numbers are not

```
a=6, b=3
add //method operates addition
c=9
sub
d=3
mul //method operates multiplication
e=18
div //method operates division
f=2
```

**Fig. 12.** Output result after dynamic analysis

```
static void dummy(int p){
    int x=(int)(Math.random()*100);
    if(((2*x+1)%2)==0){
        //condition which is always false
        method4();}
    else{p=2*(p+3)-p+6;}
    //dummy operation for p
    switchfunc(p);}
```

**Fig. 13.** Outline of dummy method

**Table 1.** Measurement result of program execution time

Source code	Frequency of random Number generation	Execution time (10 <sup>-6</sup> s)	Increase rate of Execution time(%)
P0	-	605	-
P1	8	619	2
P2	40	627	4
P0'	-	725	-
P1'	10	736	2
P2'	50	744	3
P0''	-	2,574	-
P1''	10	3,000	17
P2''	50	3,000	17

the same in each execution, so we gave a number sequence consists of probable value which will be generated during execution. Table 1 shows the result. The experimental environment is as follows.

- Processor: Intel Pentium III, 1GHz
- Memory: 512MB RAM
- Windows 2000 Service Pack 4
- j2sdk-1\_4\_2\_06-windows-i586-p.exe

From table 1, the difference of the execution time between obfuscated programs and original program is less than 1/1000 seconds. Thus, we can say the frequency of random number generation has a little influence to the execution efficiency, and execution time between original program and obfuscated program by our scheme.

## 6 Conclusion

In this paper, we introduced software obfuscation scheme using random numbers. We explained how to obfuscate control flow and extended this scheme to apply to the program having complicate control flow.

After introducing our obfuscation scheme, we studied about the scheme. In our scheme, the more the method number is, the bigger the cost of static analysis becomes. And we confirmed the influence of random number generation on execution efficiency is small. We found out that our scheme is vulnerable to dynamic analysis, then explained a countermeasure that inserting dummy method which has no influence on program execution result. In the future work, we consider scheme to make distinguishing dummy method and original method difficult, and study evaluation about quantitative security analysis of proposed scheme.

## References

1. Akito Monden, Yoshihiro Takada, Kouji Torii, "Methods for Scrambling Programs Containing Loops," IEICE Trans. D-I Vol.J80-D-I No.7 pp.1-11, July 1997.
2. T. Ogiso, Y. Sakabe, M. Soushi, "Software obfuscation on a theoretical basis and its implementation," IEICE Trans. Fundamentals, Vol.E86-A,No.1, pp.176-186, 2003.
3. Kazuhide Fukushima, Toshihiro Tabata, Kouichi Sakurai, "Proposal and Evaluation of Obfuscation Scheme for Java Source Codes by Partial Destruction of Encapsulation," Proc. of International Symposium on Information Science and Electrical Engineering 2003 (ISEE 2003), pp.389-392 (11, 2003).
4. Hirotsugu Sato, Akito Monden, Ken-ichi Matsumoto, "Program Obfuscation by Coding Data and Its Operation," Technical Report of IEICE, Technical Group on Information Theory, Vol. IT2002-49, pp.13-18, Mar. 2002.
5. D. Low, "Java control flow obfuscation", Master of Science Thesis, Department of Computer Science, The University of Auckland, 1998.

# Authenticated Public Key Distribution Scheme Without Trusted Third Party\*

Jae Hyung Koo, Bum Han Kim, and Dong Hoon Lee

Center for Information Security Technologies(CIST),  
Korea University, Seoul, Korea  
{ideao, anewholic}@cist.korea.ac.kr  
donghlee@korea.ac.kr

**Abstract.** Public key authentication is necessary to prevent a valid public key of a user from being compromised by a malicious user. Namely, if it is not provided, an adversary can read all encrypted messages between a sender and a receiver by substituting the public key of the receiver with her public key. In general, a certificate issued from and digitally signed by a publicly trusted certificate authority (CA) guarantees public key authentication under the assumption that all users can get the public key of the CA to verify the validity of certificates, *i.e.*, the signatures of the CA. The assumption is practical and widely used in the real world. However, if the CA is down by a system faults or destroyed by a terror or a war, the assumption can not be preserved. In this paper, we propose a simple and practical scheme for public key authentication without any trusted third party. The scheme basically uses a message authentication code (MAC) taking a short random value as a key to authenticate the exchanged public keys. Our scheme also can be adopted in the environments such as ad-hoc or ubiquitous in which it is hard to settle a publicly trusted authority.

**Keywords:** Key Management and Authentication, Public-key Cryptography, Public Key Infrastructure (PKI).

## 1 Introduction

Public key authentication is a method to confirm whether the received public key really belongs to the communication partner or not. If it is not provided, all public key algorithms become vulnerable to a key substitution attack. For example, an adversary who tries to eavesdrop the communication from *Alice* to *Bob* can easily obtain all messages by substituting the *Bob's* public key with her public key. Public key infrastructure (PKI) [19] is very popular technique to authenticate all the public keys of registered users. In PKI, there exists a publicly trusted third party, called certificate authority (CA) which guarantees

---

\* This work was supported (in part) by the Ministry of Information&Communications, Korea, under the Information Technology Research Center (ITRC) Support Program.

the validity of users' public keys. As a method, it issues a digital certificate for each user which contains user's information (*e.g.*, name, e-mail address and so on), user's public key and CA's signature guaranteeing the correctness of them. So, each user can easily check that the public key included in the certificate is the actual public key of whom he/she wants to communicate with by verifying the CA's signature. In fact, a method for the authentication of the CA's public key (*i.e.*, verification key for the signatures) should be provided. Otherwise, an adversary may impersonate the CA and issue certificates of users as though she is the CA. Therefore, every users must verify the public key of the CA whenever they try to start a communication with each other and it is obviously costly. Moreover, if users keep certificates issued from different CAs (there are lots of CAs in the world to serve huge numbers of people), there should be a mechanism to verify the certificates from different CAs. In general, CAs exchange the certificates of their public keys to prove the validity of their public keys and issue new certificates for other CAs to enable their registered users to verify the certificates made by the CAs.

As we explained, the role of CAs in PKI is very important. That means, if they are shut down or destroyed by disaster such as an earthquake, a terror or a war, public key authentication is no longer guaranteed. To overcome the problems especially caused by the disabled CAs, several mechanisms such as password based schemes (PBS) [4, 5, 14, 20, 21, 24, 25] and ID based schemes (IBS) [3, 13, 16] have been studied. However, they also assume the existence of trusted authorities such as a server in PBS and a private key generator (PKG) in IBS.

PBS basically considers a client-server environment. So, users register their passwords into a server and the server keeps the passwords in a secure database. The users and the server use the passwords to authenticate each other. If the server is not a trusted entity, all of the users' passwords registered in it can be revealed. There are variants of PBS [10, 9] in which users with different passwords establish a session key by the help of a server. Namely, users authenticate themselves to a server by proving the knowledge of their passwords. The server gives a method to the users so that they can exchange messages to build a session key with authentication. In this case, if the server is down, then they can not share a key. Several schemes considered an authenticated key exchange in the ad-hoc environments [1, 17, 23] where there is no trusted authority. However, they have several security breaches because of the weak password. For example, the scheme in [1] uses a password as a key for a symmetric encryption algorithm so it might be exposed while users are communicating with each other (the password may be revealed before they start a communication by eavesdropping when they share a password). With the exposed password, an adversary can impersonate a user.

In IBS, a PKG plays a role of a trusted key distribution center. So, it generates users' private keys for their public keys which they select. The public keys can be their e-mail addresses or other public but short information. Basically, whenever *Alice* wants to send a secret information to *Bob*, she can use public key encryption with taking his e-mail address as his public key. Since, *Alice* uses *Bob's* e-mail address as his public key, public key authentication can be easily

guaranteed (it is not so hard to confirm the correctness of the *Bob's* e-mail address). However, the PKG always knows all the secret keys of users. Moreover, if the PKG is down, then it is impossible for users to obtain new private keys when they update their public information.

As we mentioned, currently researched public key authentication mechanisms have a potential security breach for the disabled trusted third party. Therefore, we should consider the authenticated public key exchange without any assistance of the trusted third party.

Our contributions are two folds:

- We first consider a one-time password keyed message authentication code (OPK-MAC) to provide the public key authentication without any help of a publicly trusted authority. Since the one-time password is only used to build a temporary MAC key, we do not need to consider the off-line password guessing attack which frequently occurs in password based key exchange schemes [4, 20].
- Our idea is a generic scheme so it can be adopted in any public key based key exchange scheme. In addition, by using our scheme, we can remove the costly certificate (*i.e.*, digital signature) based public key authentication.

The structure of this paper is as follows : in Section 2 we take a look at several related works and in Section 3 we explain the notions concerned with our scheme. Section 4 describes our scheme and we conclude in Section 5. Because of the lack of pages, we omit the security proof and applications which our scheme can be adopted. However, they will be provided in the final (full) paper.

## 2 Related Works

Public key cryptography is widely used in the real world. However, it relies on infrastructure, called public key infrastructure (PKI) with online and publicly trusted certificate authority (CA) [19]. For example, in secure socket layer (SSL) [27] which is well-known security technique on web (*i.e.*, internet), the correctness of the public keys of users are guaranteed by certificates issued from CA(s). Therefore, the entire security of PKI depends on the security of CA. If CA is destroyed, then it is impossible to preserve the security of PKI, *e.g.*, there is no way to authenticate public keys.

There exist methods which can be used when PKI is not available. We review several schemes on password authenticated key exchange (PAKE) and ID based cryptography.

So far, lots of papers have been presented where a shared password among users is used to authenticate each other [4, 20, 5, 9, 10, 14, 15, 8, 24, 1]. We can categorize the schemes into three cases according to the communication topologies; communications from i) a user to a server, ii) a user to a user via a server and iii) a user to a user. Almost all of PAKE schemes have focused on the first case [4, 20, 5, 14, 15, 8, 24]. Each user registers his/her password on a server. Whenever the user and the server communicate, they can build a shared key with



authenticating each other with the password. So, the user needs not to check any complex public information such as a public key. In the second case, users share passwords with a server can communicate by the help of the server [9, 10]. The entire architecture of this case is similar to the well-known Kerberos [22]. Obviously, the first and the second cases rely on the trusted server. Namely, unless the server works honestly, then the security on user side can not be guaranteed. The last case does not require any trusted party [1]. Asokan *et. al.* [1] proposed a key agreement scheme in the ad-hoc environment under the assumption that all users share a password. They use the password as a symmetric encryption key. As a contribution, their scheme does not require a trusted third party but it may be weak for a password guessing attack by simultaneously executing an order checking attack.

ID based cryptography is another approach to make encryption or signature schemes free from the trusted third party (TTP). The basic concept of ID based scheme proposed by Shamir [28] is to use users' public information such as their e-mail addresses or IP addresses as the public keys of them. By adopting the ID based cryptography, users can eliminate the computation and communication to verify the certificates in PKI. With the merits, lots of schemes have been proposed so far [3, 13, 16]. In the ID based scheme, a user chooses his public information (*i.e.*, e-mail address) to be used as his/her public key and sends it to a private key generator (PKG). After the PKG checks the user's identity, it generates a private key associated with the public information and sends the key to the user through a secure channel. When *Alice* tries to send a message secretly to *Bob*, she can encrypt the message by using the public information of *Bob* and the public key of the PKG. *Bob*, then, decrypts the encrypted message by using his private key. The concept of ID based scheme is very useful and there are many papers dealing with encryption [6, 7] and signature [11].

However, ID based scheme has several drawbacks: the PKG generates all the private keys of the users (so the PKG knows the keys) and there should be a method to construct a secure channel when the users receive their private keys from the PKG. We point out that if the PKG does not work by a system fault or attacks such as a system hacking, then no user can obtain his/her private key for his/her public information. Moreover, establishing a secure channel for the complex and long information (*i.e.*, private key) is very hard. Therefore, a way to generate private keys by themselves is strongly required.

### 3 Preliminaries

We briefly introduce several basic schemes concerned with our scheme.

#### - One-time Password and Message Authentication Code (MAC)

In general, password based schemes use long-lived passwords in a client-server setting. So, a user registers his/her password in a server and uses the password when he/she logs in the server with authentication. Since a password is very weak information with low entropy, password based schemes basically should be

strong against a dictionary attack. To prevent the dictionary attack, password based schemes require additional cost for computation and communication to conceal the password securely in the exchanged messages to achieve some cryptographic goal. Mostly, password based schemes use the password to authenticate the identities of communicating users to build a session key. Therefore, a revealed password may affect other sessions. Namely, an adversary with the password and the captured messages from previous sessions can obtain the information of session keys built in the sessions. If we use the password to authenticate the public keys, public key authentication can not be preserved because an adversary may switch a user's public key with her public key and can pass the authentication phase by using the exposed password. A good solution to avoid the dictionary attack is to use a randomized (*i.e.*, one-time) password. Since, a user chooses different passwords in different sessions, even if an adversary finds out a valid password, he/she can not get any information of messages transmitted in different sessions.

Message authentication code (MAC) such as AES-CBC-MAC [2, 12] is used to give an integrity for a message. To use the MAC algorithm, communicating users should share a key, called MAC key. MAC is useful but key setup among users is very difficult. A user communicates with  $n$  users has to keep  $n$  keys (one key per a user). An easy method to overcome the hardness of sharing MAC keys is to adopt passwords as MAC keys. Two works for international standard, rfc2898 [26] and rfc2510 [18] provide a password based MAC (PBM) in which a password with a random value, called salt is used to make a MAC key. However, to derive a MAC key, there should be a method to share a salt. If PKI is broken, it is hard for users to exchange the salt. If we do not use the salt, then the MAC key becomes vulnerable to a dictionary attack. In our scheme, we also use passwords to derive MAC keys but the passwords are one-time passwords. Since, the purpose of using MAC is not to conceal a transmitted message but to guard it from modified by an adversary, if we use one-time passwords, then an exposed password after a session is not helpful for the adversary. Namely, with the obtained password, the adversary can not modify the messages in other sessions. The only security consideration is to protect the password used to derive a MAC key from revealed to an adversary before a sender makes a MAC and sends it to a receiver. To provide the security against password exposure before sending a MAC, we use a *post-shared password* which is explained in *Remark 1* in Section 4.

## 4 Authenticated Public Key Distribution Scheme Without Trusted Third Party

In our scheme, several mathematical notions are adopted and we define them as follows. Let  $p, q$  be primes such that  $q|p-1$  and  $\mathcal{G}$  be a subgroup of  $Z_p^*$  with order  $q$ .  $g$  is a generator of  $\mathcal{G}$ . The private/public key pairs of users are set as  $x_i$  and  $y_i = g^{x_i} \bmod p$  respectively where  $1 \leq i \leq n$  and  $n$  is the number of users. In fact, the private key should be managed in a secure device such as a smartcard.

We adopt a secure message authentication code (MAC) to provide public key authentication and a function  $h(\cdot)$  mapping a input value into a point of the candidate key space of MAC. We note that  $h(\cdot)$  is a collision-resistant function so the probability of finding two input values leading a same output is negligible.

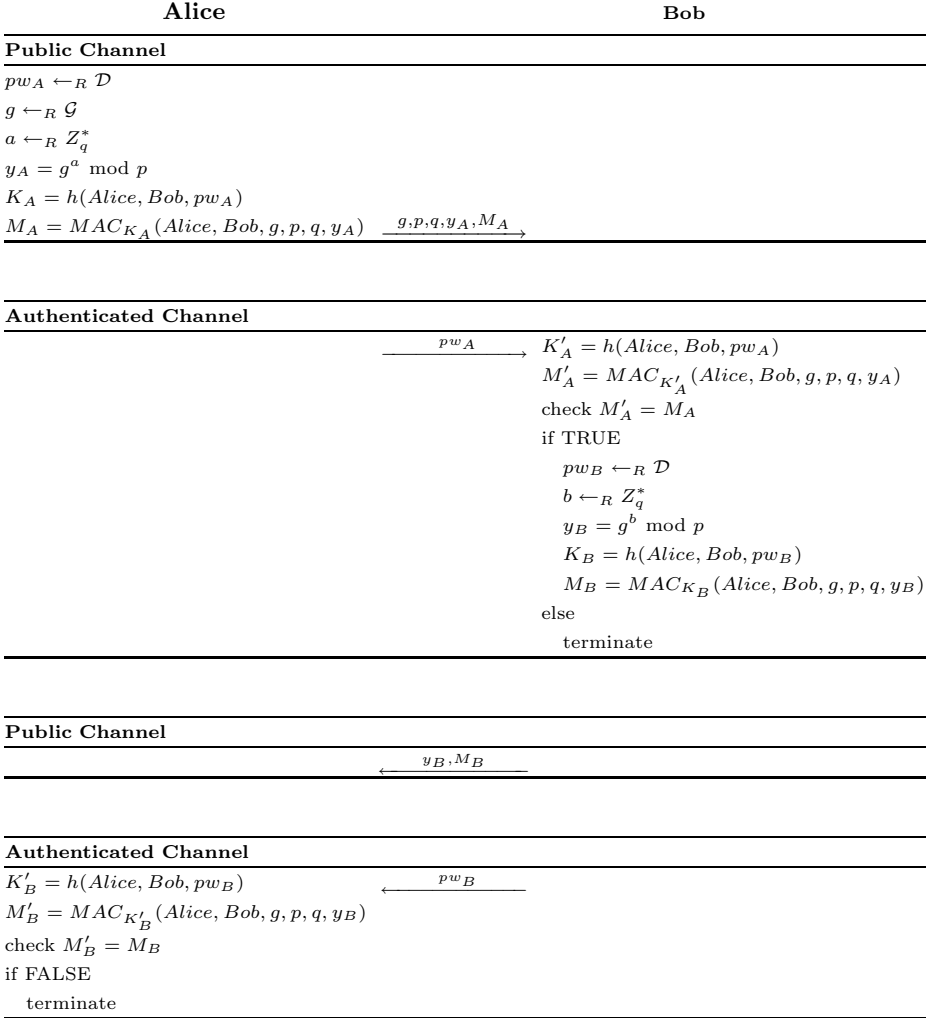
Basically, we assume that there exists an authenticated channel in which communicating users can easily check the validity of exchanged messages where the length of the messages is quiet short. For example, if *Alice* talks with her friend, *Bob* through a telephone, then she can say a few short words to *Bob* with proving that she is *Alice* and the words are exactly what she said. Namely, *Bob* can easily recognize the voice of *Alice* or some events they have shared. In our setting, *Alice* may be able to hand over her temporary password to *Bob* through phone-call. The main idea of our scheme is to send the password used to establish a MAC key after transmitting MAC.

Our scheme runs as follows (Fig. 1 also describes our scheme):

1. *Alice* picks a random password  $pw_A$  from the password dictionary  $\mathcal{D}$ .  $pw_A$  may consist of alphabetic characters (52 characters : 'a' to 'z' and 'A' to 'Z') and numeric characters (10 characters : '0' to '9'). The length of  $pw_A$  is flexible so *Alice* can choose any size of passwords. 8 character-length is sufficient because the probability of guessing a password becomes approximately  $(\frac{1}{62})^8 (= \frac{1}{218340105584896} \approx \frac{1}{2 \times (10)^{14}})$ .
2. *Alice* also selects a generator  $g$  randomly from  $\mathcal{G}$  and a random value  $a$  from  $Z_q^*$ . She computes  $y_A = g^a \bmod p$ . We note that the values,  $a$  and  $y_A$  can be both an actual private/public key pair or a temporary private/public key pair.
3. *Alice* sets a MAC key  $K_A = h(\textit{Alice}, \textit{Bob}, pw_A)$  and makes a MAC of generated values, *i.e.*,  $M_A = MAC_{K_A}(\textit{Alice}, \textit{Bob}, g, p, q, y_A)$ .
4. Finally, *Alice* transfers  $g, p, q, y_A, M_A$  to *Bob* through a general communication channel such as internet.
5. A few seconds later, *Alice* transmits  $pw_A$  through an authenticated channel. As a method, she can call *Bob* and tell *Bob*  $pw_A$  on a telephone.
6. With received  $pw_A$ , *Bob* reconstructs a MAC key  $K'_A = h(\textit{Alice}, \textit{Bob}, pw_A)$  and checks the validity of  $M_A$  by comparing it with  $MAC_{K'_A}(\textit{Alice}, \textit{Bob}, g, p, q, y_A)$ .
7. If the values are same, *Bob* selects a random password  $pw_B$  ( $pw_B \neq pw_A$ ) from  $\mathcal{D}$  and a random value  $b$  from  $Z_q^*$ . He computes public value  $y_B = g^b \bmod p$ .
8. After establishing a MAC key  $K_B = h(\textit{Alice}, \textit{Bob}, pw_B)$ , he makes a MAC,  $M_B = MAC_{K_B}(\textit{Alice}, \textit{Bob}, g, p, q, y_B)$  to send it with  $y_B$  to *Alice*.
9. As *Alice* did, he transfers  $pw_B$  through an authenticated channel.
10. *Alice* rebuilds  $K'_B = h(\textit{Alice}, \textit{Bob}, pw_B)$  and checks the validity of  $M_B$ . If  $M_B = M'_B = MAC_{K'_B}(\textit{Alice}, \textit{Bob}, g, p, q, y_B)$ , then she accepts  $y_B$  and regards that she and *Bob* share the same parameters,  $g, p, q$  correctly.

*Remark 1 (Password : pre-sharing v.s. post-sharing).*

Password based schemes assume that the communicating users share a password. So, there should be a initialization phase to setup the password. With the



**Fig. 1.** Proposed protocol

password, the users authenticate themselves to their communication partners by using the password as a symmetric key or adding the password into some operations for public key exchange or session key establishment. We denote the password as *pre-shared password*. Since the password has very low entropy, pre-shared password based schemes must consider the security against a dictionary attack. In particular, the password should not be revealed during exchanging the public keys or building a session key to prevent an adversary from an impersonation attack. In our scheme, users share a password after they exchange their public key and MACs of the keys. We denote the password as *post-shared password*. Since a sender transmits his/her password through an authenticated

channel in a few seconds after sending the public key and MAC, it is very hard for an adversary to modify the transmitted messages. Namely, the adversary should find out the password used to make a MAC key in a few second to switch the messages from a sender to a receiver with the messages which she made to impersonate the sender. We note that if we adopt an authenticated channel in pre-shared password schemes, an adversary can impersonate a user by eavesdropping the authenticated channel to obtain a password. For example, if *Alice* and *Bob* use a phone-call as an authenticated channel, then an adversary standing around *Alice* can hear her voice when she tells *Bob* her one-time password.

*Remark 2 (Authenticated Channel).*

In our scheme, we assume an authenticated channel and it is not strong assumption. In the real world, we have many methods to authenticate communication partners. A telephone is a good equipment as an authenticated channel. If *Alice* knows *Bob* well, then she can easily distinguish *Bob*'s voice from others' voices. They can also use television telephone system to authenticate each other. If they are in a same room, then they can exchange papers containing their passwords. We note that we only use the authenticated channel to share a password. Obviously, it is very hard to read or say the long and complex strings, *i.e.*, a public key, but it is very easy for the short password. So, using an authenticated channel to share a password is practical.

## 5 Efficiency

Table 1 shows the comparison results of our scheme with existing authentication schemes. In the table, *Cost* means the computation and communication cost to authenticate the communicating users and exchanged messages to achieve a cryptographic goal such as public key authentication and session key establishment. TTP denotes the trusted third party. We also present the role of each TTP and we can easily recognize that all of the schemes excluding our scheme do not work without help of TTP.

**Table 1.** comparison results of our scheme with other authentication mechanisms

Protocols	<i>Cost for authentication</i>	<i>TTP</i>
PKI	very high	CA
PBS	low (knowledge of password check)	Server
IBS	very low (public information validity check)	PKG
Proposed scheme	very low (two MAC operations)	None

Because of the lack of the pages, we only provide the brief comparison results. More detailed comparisons with other schemes will be shown in the full paper.

## 6 Conclusion

We pointed out that public key authentication especially through PKI can not be preserved if a CA is shut down or destroyed. To prepare the such situations, we proposed a practical solution for public key authentication by setting an authentication channel and adopting a MAC with post-shared one-time password. Our idea is a generic scheme so it can be easily adopted in all the environments where there is no trusted third party such as CA. Therefore, our scheme is very applicable and it can be also used for the key agreement in the ad-hoc and ubiquitous environments.

## References

1. N. Asokan and P. Ginzboorg. Key-Agreement in Ad-hoc Networks. In *Forth Nordic Workshop on Secure Computer Systems*, 1999.
2. "Advanced Encryption Standard". Available at "<http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>".
3. J. Baek and Y. Zheng. Identity-based Threshold Decryption. In *PKC 2004*, LNCS 2947, pp. 262-276, Springer-Verlag, 2004.
4. M. Bellare, D. Pointcheval and P. Rogaway. Authenticated Key Exchange Secure against Dictionary Attacks. In *Eurocrypt 2000*, LNCS 1807, pp. 139-155, Springer-Verlag, 2000.
5. V. Bokyo, P. Mackenzie, and S. Patel. Provably Secure Password-Authenticated Key Exchange using Diffie-Hellman. In *Eurocrypt 2000*, LNCS 1807, Springer-Verlag, 2000.
6. D. Boneh and X. Boyen. Efficient Selective-ID Secure Identity based Encryption. In *Eurocrypt 2004*, LNCS 3027, pp. 223-238, Springer-Verlag, 2004.
7. J. Boneh and M. Franklin. Identity-based Encryption from the Weil Pairing. In *Crypto 2001*, LNCS 2139, pp. 213-229, Springer-Verlag, 2001.
8. M. Boyarsky. Public-Key Cryptography and Password Protocols: The Multi-User Case. In *ACM CCCS '99*, 1999.
9. J. Byun and D. Lee. N-Party Encrypted Diffie-Hellman Key Exchange Using Different Passwords. In *ACNS 2005*, LNCS 3531, pp. 75-90, Springer-Verlag, 2005.
10. J. Byun, I. Jeong, D. Lee and C. Park. Password-Authenticated Key Exchange between Clients with Different Passwords. In *ICICS 02*, LNCS 2513, pp. 134-146, Springer-Verlag, 2002.
11. J. Cha and J. Cheon. An Identity-based Signature from Diffie-Hellman Groups. In *PKC 2003*, LNCS 2567, pp. 18-30, Springer-Verlag, 2003.
12. "DES modes of operation". Available at "<http://www.itl.nist.gov/fipspubs/fip81.htm>".
13. C. Gentry and A. Silverberg. Hierarchical ID-based Cryptography. In *Asiacrypt 2002*, LNCS 2501, Springer-Verlag, 2002.
14. O. Goldreich and Y. Lindell. Session Key Generation using Human Passwords Only. In *CRYPTO 2001*, Springer-Verlag pp.408-432, 2001.
15. S. Halevi and H. Krawczyk. Public Key Cryptography and Password Protocols. *ACM Transactions on Information and System Security*, 1999.
16. J. Horwitz and B. Lynn. Toward Hierarchical Identity-based Encryption. In *Eurocrypt 2002*, LNCS 2332, Springer-Verlag, 2002.

17. J. Hubaux, L. Buttyan and S. Capkun. The Quest for Security in Mobile Ad-hoc Networks. In *MOBIHOC 2001*, pp. 146-155, 2001.
18. "Internet X.509 Public Key Infrastructure-Certificate Management Protocols". Available at "<http://www.ietf.org/rfc/rfc2510.txt>".
19. "Internet X.509 Public Key Infrastructure-Certificate and Certificate Revocation List (CRL) Profile". Available at "<http://www.ietf.org/rfc/rfc3280.txt>".
20. J. Katz, R. Ostrovsky and M. Yung. Efficient Password-Authenticated Key Exchange Using Human-Memorable Passwords. In *Eurocrypt 2001*, LNCS 2045, pp.475-494, Springer-Verlag, 2001.
21. K. Kobara and H. Imai. Pretty-Simple Password-Authenticated Key-Exchange Under Standard Assumptions. *IEICE Trans.*, vol. E85-A, no. 10, pp. 2229-2237, 2002.
22. J. Kohl and B. Neuman. The Kerberos Network Authentication Service (V5). RFC 1510, 1993..
23. J. Kong, P. Zerfos, H. Luo, S. Lu and L. Zhang. Providing robust and ubiquitous security support for mobile ad-hoc networks. In *IEEE International Conference on Network Protocols*, 2001.
24. T. Kwon. Practical Authenticated Key Agreement using Passwords. In *Information Security, 7th International Conference, ISC 2004*, LNCS 3225, Springer-Verlag, 2004.
25. P. MacKenzie and R. Swaminathan. Secure Network Authentication with Password Identification. IEEE P1363a, 1999.
26. "RFC 2898 - PKCS #5: Password-Based Cryptography Specification Version 2.0". Available at "<http://www.faqs.org/rfcs/rfc2898.html>".
27. Secure Socket Layer (SSL). Available at "<http://openssl.org>".
28. A. Shamir. Identity-based Cryptosystems. Crypto'84, 1984.

# Cryptanalysis of a Generalized Anonymous Buyer-Seller Watermarking Protocol of IWDW 2004

Bok-Min Goi<sup>1</sup>, Raphael C.-W. Phan<sup>2</sup>, and M.U. Siddiqi<sup>1,\*</sup>

<sup>1</sup> Centre for Cryptography and Information Security (CCIS),  
Faculty of Engineering, Multimedia University,  
63100 Cyberjaya, Malaysia  
bmgoi@mmu.edu.my

<sup>2</sup> Information Security Research (iSECURES) Lab,  
Swinburne University of Technology (Sarawak Campus),  
93576 Kuching, Malaysia  
rphan@swinburne.edu.my

**Abstract.** In this paper, we analyze the security of a generalized anonymous buyer-seller watermarking protocol recently proposed by Choi and Park at IWDW 2004. We prove that it has not met the designers' intended security criteria by showing that an attacker can actually: (1) discover the unique buyer's watermark which was chosen by the watermark certificate center (*WCC*), and (2) decrypt the encrypted watermarked digital content without any extra cost. Also, it is surprising to note that when designing their protocol, the designers did not take into consideration the conspiracy attacks.

**Keywords:** Digital Watermarking, Buyer-Seller Watermarking Protocol, Cryptanalysis, Copyright Protection.

## 1 Introduction

In this information age, all types of multimedia information are being stored and processed in digital form, because of many advantages over the traditional analog counterpart. Unfortunately, since the duplication of digital multimedia content results in perfectly identical copies, many multimedia content providers are hesitant to sell/distribute their content digitally. *Digital watermarking* [15, 5] and *digital fingerprinting* [12, 13] are mainly designed to overcome this problem—the *copyright protection issue*. A *buyer-seller watermarking protocol* is a combination of both; more precisely, it allows to trace redistribution of the digital contents by extracting the original buyer's information (fingerprint) and to prove the content owner by extracting the sellers' information (watermark) from the redistributed contents. Therefore, it does protect the rights and interests of not only the seller

---

\* The first author acknowledges the Malaysia IRPA grant (04-99-01-00003-EAR).



but also of the buyer. In the literature, many buyer-seller watermarking protocols have been proposed [1-4, 7-10].

In this paper, we analyze the security of a generalized anonymous buyer-seller watermarking protocol recently proposed by Choi and Park at IWDW 2004 [3]. At first glance, the new protocol looks quite attractive because it can be generalized to *multi-purchase environments* where a set of buyers can purchase multiple distinct contents from a set of sellers, and this solves the open problem stated in Goi *et al.* [7]. However, we show that it does not achieve the security criteria set out by the designers, namely that an attacker can actually: (1) discover the unique buyer watermark chosen by the watermark certificate center (*WCC*), and (2) decrypt and obtain the encrypted watermarked digital content. It is also surprising that when designing the protocol, the designers did not consider security against *conspiracy attacks* when in fact even Memon and Wong [10] – the designers of the original buyer-seller watermarking protocol – had commented that it is undesirable to place complete trust on a single source, including the watermark certification center, and that such attacks had already been considered extensively in this literature [8, 3, 7]. To solve the conspiracy attack, we refer readers to Goi *et al.* [7] where the buyer should generate his own private watermark and convince the certificate authority that s/he owns the secret watermark, via the zero-knowledge proof protocol.

## 1.1 Related Work

In 1998, Qiao and Nahrstedt [14] presented an *owner-customer* watermarking protocol that solved the problem of rightful ownership. Unfortunately, it is a *symmetric scheme* and does not guarantee the buyer's security. The first truly buyer-seller watermarking protocol that withheld the buyer's unique watermark (fingerprint) from the seller was proposed by Memon and Wong [10]. The Memon-Wong protocol is an *asymmetric scheme* where even the seller is not able to reproduce the unique watermarked content. Since then, several variants of this protocol have been proposed in the literature, including the Chang-Chung protocol and Cheung *et al.* protocol which were proposed at ICCT '03 [1] and HICSS '04 [2], respectively. Also worth mentioning is the Lei *et al.* protocol [9] which is based on the unbinding problem.

Independently of these, the issue of the importance of "anonymity" was raised by Ju *et al.* at ICISC '02 [8] when they presented an anonymous version of the Memon-Wong protocol. Choi *et al.* later at ACNS '03 [3] presented conspiracy attacks on the Ju *et al.* variant and further modified it such that it resists these attacks. However, Goi *et al.* at ACNS '04 [7] showed that the Choi *et al.* protocol was still insecure against conspiracy attacks and also can not provide *full anonymity*. They then proposed a variant that achieves full anonymity. Finally, Choi and Park at IWDW '04 generalized on their earlier ACNS '03 work of [3] to multi-purchase environments. They also claimed that their protocol can be implemented for mobile communication by employing mobile agents with an extra step - *delegation* step. A comparison of security features and primitive

requirements of various proposed buyer-seller watermarking protocols is shown in Table 1.

**Table 1.** Comparison of Security Features and Primitive Requirements of Various Buyer-Seller Watermarking Protocols

		Memon 01 [10]	Ju 02 [8]	Chang 03 [1]	Choi 03 [3]	Cheung 04 [2]	Goi 04 [7]	Lei 04 [9]	Choi 04 [3]
Services & Security Provided	Anonymity	×	√	×	√	×	√	√	√
	Conspiracy attacks	×	×	×	×	×	√	×	×
	Multi-purchase	×	×	×	×	×	×	×	√
Primitive Requirements	Homomorphic †	√	√	×	√	√	√	×	√
	Commutative ‡	×	×	×	√	√	×	×	×
	Linearity †	√	√	√	√	×	√	×	√

† : Homomorphic encryption algorithm and linear watermarking scheme are defined in subsection 2.2.

‡ : An encryption algorithm is said to be *commutative*, if for a multiple encrypted (decrypted) message the same resultant ciphertext (plaintext) will be obtained irrespective of its the order of encryption [7].

### 1.2 Security Criteria

We describe in this subsection some security criteria that are expected to be achieved by a sound anonymous buyer-seller watermarking protocol, as follows [7, 3]:

- **Anonymity.** A buyer’s identity is protected, unless s/he is found guilty.
- **Unlinkability.** Nobody is able to determine whether the different watermarked contents are purchased by the same buyer.
- **Traceability.** The buyer who has illegally redistributed watermarked contents can be traced.
- **No-Framing.** Nobody can accuse an honest buyer.
- **Non-Repudiation.** The guilty buyer cannot deny that the unauthorized copies of the content were not created by him.

Obviously, the security of a seller-buyer watermarking protocol is dependent on the underlying watermarking scheme. Hence, the used watermarking scheme must be collusion tolerant; more precisely, nobody can find and delete the embedded watermark (invisible type) from the content without knowing the watermark.

## 2 Preliminaries

### 2.1 Notations

For ease of explanation, we stick to the notations used in [3] as follows:

---

$S$	the seller who sells the digital multimedia content
$B$	the buyer who can buy watermarked contents anonymously
$RC$	the agent who can verify the buyer's identity and issue the buyer anonymous certificates, $Cert(y_1)$
$WCC$	the agent who can issue watermarks to buyers upon request and certify them
$CA$	certification authority who can issue the certificate and a pair of keys $(x, y)$ for every agent in the PKI
$X$	original content with $m$ elements $x_1, x_2, \dots, x_m$
$W$	watermark with $n$ elements $w_1, w_2, \dots, w_n$ generated by $WCC$ , where $n \leq m$
$V$	watermark with $n$ elements $v_1, v_2, \dots, v_n$ generated by the seller, where $n \leq m$
$X', X''$	watermarked content
$X \otimes W$	embed watermark, $W$ into $X$ with insertion operation, $\otimes$
$\sigma$	random permutation function chosen by $S$
$t$	total number of contents to be purchased
$text$	valid set of operations that the agent is allowed to perform while using the certificates
$\parallel$	string concatenation

---

## 2.2 Building Blocks

The protocols discussed here use *public key cryptography* [11]; each agent,  $A$  possesses a pair of keys: public key,  $y_A$  and private key  $x_A$  – which are obtained from a certificate authority center ( $CA$ ). For convenience, we stick to  $y_A \equiv g^{x_A} \pmod p$ , where  $p$  is a large prime such that  $\frac{p-1}{2}$  is also a prime, and  $g$  is a generator of the multiplication group,  $Z_p$ . All arithmetic operations are performed under  $Z_p$ , unless otherwise specified. We denote  $E_y(m)$  to mean the message,  $m$  encrypted with the public key,  $y$ . Any agent can encrypt a message for  $A$  using  $y_A$ , but only  $A$  can decrypt this message with  $x_A$ :  $D_{x_A}[E_{y_A}(m)]$ . Furthermore,  $A$  can sign a message by encrypting it with  $x_A$ , denoted as  $sign_{x_A}(m)$ . We assume that all agents have registered with the  $CA$  beforehand and have their own pair of keys: the seller,  $S$  with  $(x_S, y_S)$ , the buyer,  $B$  with  $(x_B, y_B)$ ,  $RC$  with  $(x_R, y_R)$  and  $WCC$  with  $(x_W, y_W)$ . Also, the public-key encryption algorithm used in this paper must be homomorphic. A well-known homomorphic encryption algorithm is the RSA with respect to the multiplication operation.

Besides robustness, in terms of various digital processing operations, printing and re-scanning, and collusion attacks, the underlying watermarking scheme adopted must have *linearity* property. This is because the seller needs to permute the original watermark generated by  $WCC$  with a random permutation function,  $\sigma$  before embedding it into the content. The detailed watermark embedding process is described as follows:

$$\begin{aligned}
 X' &= X \otimes \sigma(W) \\
 &= (x_1, x_2, \dots, x_m) \otimes \sigma(w_1, w_2, \dots, w_n) \\
 &= (x_1 \otimes w_{\sigma(1)}), (x_2 \otimes w_{\sigma(2)}), \dots, (x_n \otimes w_{\sigma(n)}), x_{n+1}, x_{n+2}, \dots, x_m
 \end{aligned}$$

Therefore, the robust Cox’s invisible watermarking algorithm [5] is a suitable choice and has intensively been used in Memon-Wong protocol and its variants.

### 3 An Overview of the Choi-Park Protocol

Basically, there are four steps involved in the Choi-Park Protocol: (1) Registration, (2) Watermarking generation, (3) Watermark insertion, and (4) Copyright violator identification.

In order to ensure that Choi-Park protocol works, several typos and mistakes in [3] have to be corrected. They are:

- **Registration step:** Firstly,  $RC$  should keep  $(y_B, y_1, cert(y_1))$  in database  $RC\_DB$ , but not  $(y_B, y_2, cert(y_1))$ . Further, although the designers did not consider this,  $x_{B_2}$  has to be stored in database  $RC\_DB$ . This is because  $x_{B_2}$  is later used during the Copyright violator identification step to prove and disclose the buyer’s identity if s/he is found guilty; i.e.,  $y_1^{x_{B_2}} = y_B$ .
- **Watermark generation step:** Note that  $(k_1, k_2, \dots, k_t) \in Z_p$  have to be chosen carefully, so that the obtained  $y_i^* = (y_1^{k_i}, g^{k_i})$  contains no zero element of  $y_1^{k_i}$ , for  $i = 1, 2, \dots, t$ . Otherwise, the encryption function:  $E_{y_i^*}(m) = (m \cdot y_1^{k_i} \| g^{k_i})$  would turn out to be trivial.
- **Watermark insertion step:** The validity of  $sign\_B_i = sign_{x_{B_1}}(text_i)$  is verified by  $S$  using  $y_1 = g^{x_{B_1}}$  (anonymous public key of  $x_{B_1}$ ) but not  $y_i^*$ . Hence,  $B$  has to send  $y_1$  to  $S$ . (This will cause the protocol to become linkable although it still provides anonymity service.) Furthermore, according to the definition of the encryption function as used in [3], the final encrypted watermarked digital content should be defined as  $E_{y_i^*}(X_i) = E_{y_i^*}(X_i \otimes V_i \otimes \sigma_i(W_i) \cdot y_1^{k_i} \| g^{k_i})$ , and not,  $E_{y_i^*}(X_i) = E_{y_i^*}(X_i \otimes V_i \otimes \sigma_i(W_i) \cdot y_i^* \| g^{k_i})$ . Therefore, the decryption function is performed as follows:

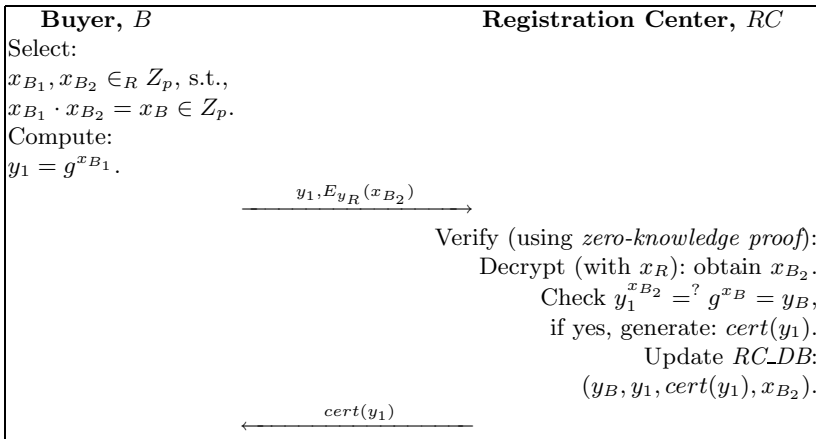
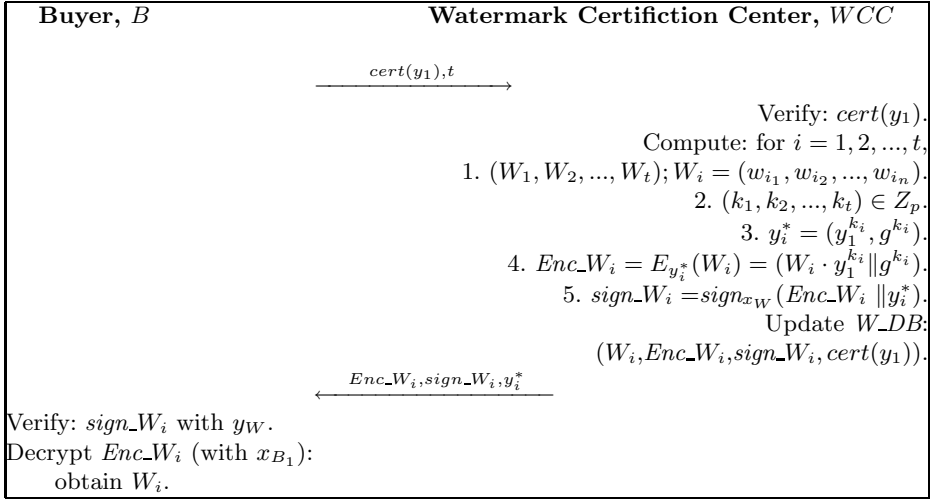
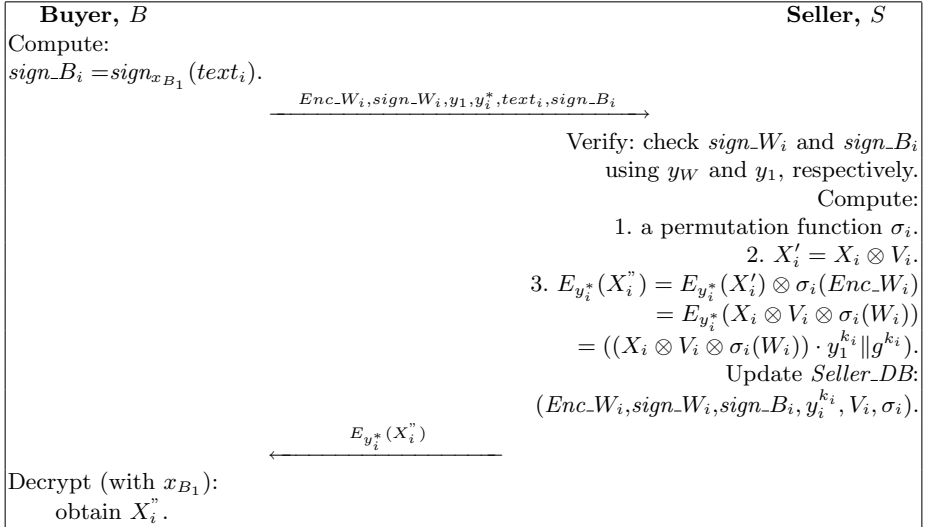


Fig. 1. Choi-Park Protocol: Registration


**Fig. 2.** Choi-Park Protocol: Watermark Generation

**Fig. 3.** Choi-Park Protocol: Watermark Insertion

$$\begin{aligned}
 D_{x_{B1}}[E_{y_i^*}(X_i'')] &= \frac{X_i \otimes V_i \otimes \sigma_i(W_i) \cdot y_1^{k_i}}{(g^{k_i})^{x_{B1}}} \\
 &= \frac{X_i \otimes V_i \otimes \sigma_i(W_i) \cdot y_1^{k_i}}{y_1^{k_i}} = X_i \otimes V_i \otimes \sigma_i(W_i) \quad (1)
 \end{aligned}$$

Choi and Park claimed here that only the buyer with the secret information,  $x_{B_1}$  can perform the decryption and then obtain the final watermarked content,  $X_i^n$ . However, we prove in Section 4 that it is not true.

For compactness of description, all these steps, except for step (4), are illustrated in Figures 1, 2 and 3, respectively. The proposed protocol is efficient because each step can be completed in one round. During the registration step, as shown in Figure 1, by providing  $y_1$  (which is used for achieving anonymity),  $B$  convinces  $RC$  of possession of  $x_{B_1}$  via a zero-knowledge proof [6]. We omit the copyright violator identification step as it is straightforward and irrelevant to our attacks and discussions in this paper. We refer interested readers to [3] for further details.

## 4 Attacking the Choi-Park Protocol

The main problem of the Choi-Park protocol is the encryption process during the watermark generation step:  $WCC$  encrypts each watermark  $W_i$  with  $y_i^* = (y_1^{k_i}, g^{k_i})$  such that,  $Enc\_W_i = E_{y_i^*}(m) = (m \cdot y_1^{k_i} || g^{k_i})$ , for  $i = 1, 2, \dots, t$ . However,  $y_i^*$ , together with  $Enc\_W_i$  and  $sign\_W_i$ , are sent in clear form, both during the watermark generation step and during the watermark insertion step. This causes a serious security flaw and the protocol becomes totally insecure. More precisely, upon receiving/intercepting these messages, the seller/ anyone can decrypt  $Enc\_W_i$  to obtain the watermark  $W_i$  even without  $k_i$ , and  $x_{B_1}$  (which are secretly kept by  $WCC$  and the buyer, respectively). This is because one can compute  $(y_1^{k_i})^{-1}$ , the inverse of  $y_1^{k_i}$  easily. For instance, the well-known extended Euclidean algorithm [11] can be used for computing multiplicative inverses in  $Z_p$ , where  $p$  is an  $n$ -bit prime integer. The computational complexity of the given algorithm is only  $O((lg p)^2)$  or  $O(n^2)$ ; in other words, it can be done in polynomial time.

Upon computing  $(y_1^{k_i})^{-1}$ , we multiply it with  $Enc\_W_i = (W_i \cdot y_1^{k_i} || g^{k_i})$  (note that doing so is equivalent to decryption) to recover  $W_i$ . This disproves the claims in [3] that only the buyer with knowledge of  $x_{B_1}$  can decrypt  $Enc\_W_i$  to get the unique watermark  $W_i$  based on Eq.(1). This causes several serious security issues, some of which are further explained in detail in following subsections.

### 4.1 Attacking the Buyer's Security

**By a Malicious Seller.** Once a malicious seller has obtained the unique buyer's watermark,  $W_i$  as described above, then with all other the necessary information, i.e.,  $X, V_i, \sigma_i$  that he has access to, he can reproduce and redistribute illegally the watermarked content for his own gain. When the illegal watermarked content is found in the market, the innocent buyer will be accused. Hence, "no-framing" and "non-repudiation" cannot be provided.

**By any Outsider.** With the knowledge of  $W_i$ , an attacker can simply embed it into a digital content which is not (never) purchased by the buyer, who may never be aware of this. Again, an innocent buyer will be accused.

## 4.2 Attacking the Seller's Security

As mentioned above, with the information  $y_1^{k_i}$ , the decryption process can be carried by anyone. Firstly, an attacker intercepts the encrypted watermarked digital content,  $E_{y_i^*}(X_i^*)$  during the watermark insertion step and then decrypts it to obtain the watermarked content,  $X_i^*$ , although s/he does not pay to the seller.

## 5 Discussions

Our attack exploited the fact that  $y_i^* = (y_1^{k_i}, g^{k_i})$  is sent in the clear, and that the encryption method used by the designers is mere multiplication with  $y_1^{k_i}$  (followed by concatenation with  $g^{k_i}$ ), and correspondingly, the decryption (see equation (1)) is simply division with the same, which translates to multiplication with the multiplicative inverse of  $y_1^{k_i}$ . Although  $x_{B_1}$  was used in the generation of  $y_1$  and can therefore be used in the decryption process, the fact is that even without  $x_{B_1}$ , decryption can still be done with  $y_1$ . This is very clear from equation (1). This is quite different from the encryption and decryption processes used in RSA-type schemes where both involved exponentiations.

Finally, it is very surprising that during the design of their protocol Choi and Park did not take into consideration of the conspiracy attacks. It is obvious that once the seller colludes with *RC* or *WCC*, the anonymity and other security services provided by the protocol will be compromised totally. In order to solve this, the buyer should be responsible to generate his/her own private watermark and convince the certificate authority that s/he owns the secret watermark, via the zero-knowledge proof protocol, as proposed by Goi *et al.* in [7].

## 6 Conclusions

In this paper, we have shown that Choi and Park protocol is flawed. This is due to the leak of  $y_i^*$  during the protocol that allows an attacker to discover the unique buyer watermark and also further decrypt the encrypted watermarked digital content. Therefore, it cannot provide “no framing” and “non-repudiation”, “unlinkability” and “untraceability” security criteria, even if the underlying watermarking scheme is secure. Furthermore, the protocol is not secure against conspiracy attacks.

## References

1. C.C. Chang and C.Y. Chung. An Enhanced Buyer-Seller Watermarking Protocol. In *Proceedings of ICCT '03*, pp. 1779-1783, 2003.
2. S.C. Cheung, H.F. Leung and C. Wang. A Commutative Encrypted Protocol for the Privacy Protection of Watermarks in Digital Contents. In *Proceedings of the 37th Hawaii International Conference on System Sciences (HICSS-37)*, January 2004.

3. J.G. Choi and J.H. Park. A Generalization of an Anonymous Buyer-Seller Watermarking Protocol and Its Application to Mobile Communications. In *Proceedings of IWDW '04*, LNCS 3304, pp. 232-243, 2005.
4. J.G. Choi, K. Sakurai, and J.H. Park. Does It Need Trusted Third Party? Design of Buyer-Seller Watermarking Protocol without Trusted Third Party. In *Proceedings of ACNS '03*, LNCS 2846, pp. 265-279, 2003.
5. I.J. Cox, J. Kilian, T. Leighton and T. Shamoon. Secure Spread Spectrum Watermarking for Images, Audio and Video. In *IEEE Trans. on Image Processing*, vol. 6, no.12, pp.1673-1678, 1997.
6. D. Chaum. An Improved Protocol for Demonstrating Possession of Discrete Logarithms and some Generalizations. In *Proceedings of Eurocrypt '87*, LNCS 307, pp. 127-141, 1987.
7. B.M. Goi, R.C.-W. Phan, Y. Yang, F. Bao, R.H. Deng and M.U. Siddiqi. Cryptanalysis of Two Anonymous Buyer-Seller Watermarking Protocols and An Improvement for True Anonymity. In *Proceedings of ACNS '04*, LNCS 3089, pp. 369-382, 2004.
8. H.S. Ju, H.J. Kim, D.H. Lee and J.I. Lim. An Anonymous Buyer-Seller Watermarking Protocol with Anonymity Control. In *Proceedings of ICISC '02*, LNCS 2587, pp. 421-432, 2002.
9. C.-L. Lei, P.-L. Yu, P.-L. Tsai and M.-H. Chan. An Efficient and Anonymous Buyer-Seller Watermarking Protocol. In *IEEE Trans. on Image Processing*, vol. 13, no. 12, pp. 1618-1626, 2004.
10. N. Memon and P.W. Wong. A Buyer-Seller Watermarking Protocol. In *IEEE Trans. on Image Processing*, vol. 10, no.4, April 2001.
11. A.J. Menezes, P.C. van Oorschot, S.A. Vanstone. Handbook of Applied Cryptography", CRC Press, 1997.
12. B. Pfitzmann and A.R. Sadeghi. Coin-Based Anonymous Fingerprinting. In *Proceedings of Eurocrypt '99*, LNCS 1592, pp. 150-164, 1999.
13. B. Pfitzmann and M. Waidner. Anonymous Fingerprinting. In *Proceedings of Eurocrypt '97*, LNCS 1233, pp. 88-102, 1997.
14. L. Qiao and K. Nahrstedt. Watermarking Schemes and Protocols for Protecting Rightful Ownership and Customer's Rights. In *J. Visual Commun. Image Representation*, vol. 9, pp. 194-210, 1998.
15. G. Voyatzis and I. Pitas. The Use of Watermarks in the Protection of Digital Multimedia Products. In *Proc. IEEE*, Vol. 87, pp. 1197-1207, July 1999.



# Efficient RFID Authentication Protocol for Ubiquitous Computing Environment\*

Eun Young Choi, Su Mi Lee, and Dong Hoon Lee

Center for Information Security Technologies(CIST),  
Korea University, 1, 5-Ka, Anam-dong, Sungbuk-ku, Seoul, 136-701, Korea  
{bluecey, smlee}@cist.korea.ac.kr, donghlee@korea.ac.kr

**Abstract.** Radio Frequency identification (RFID) will become an important technology in remotely object identification systems. However, the use of RFID tags may create new threats to the security and privacy of individuals holding RFID tags. These threats bring several problems which are information leakage of a tag, location trace of individuals and impersonation of a tag. Low-cost RFID systems have much restrictions such as the limited computing power, passive power mechanism and low storage space. Therefore, the cost of tag's computation should be considered as an important factor in low-cost RFID systems. We propose an authentication protocol, OHLCAP which requires only *one* one-way hash function operation and hence is very efficient. Furthermore, our protocol is suitable to ubiquitous computing environment.

## 1 Introduction

A Radio Frequency Identification (RFID) tag is a microchip that is capable of transmitting a unique serial number and other additional data through RF(radio frequency) signals. The goal of a RFID system is to identify objects remotely by embedding tags into the objects. For example, goods in shops can be tagged in order to provide automatic theft-detection, or to manage the goods inventory by using wireless scanning without any handwork. RFID tags are useful tools in manufacturing, supply chain management, inventory control, etc.

A RFID system is composed of three components; tag, reader and Back-end database. The characteristics of each component are as follows.

- *RFID tag* carries an object identifying data. When a tag receives a query from a reader, the tag transmits information to the reader using RF signals.
- *RFID reader* reads and re-writes the stored data in a tag. After a reader queries to a tag and receives information from the tag, the reader forwards the information to a Back-end database.
- *Back-end database* is powerful in computational capacity and manages lots of information related to each tag. Generally we assume that an adversary can

---

\* This research was supported by grant No.R01 – 2004 – 000 – 10704 – 0 from the Korea Science & Engineering Foundation.

monitor all messages transmitted in wireless communication between a reader and a tag. However in wired communication between a reader and a Back-end database, we assume that the reader can establish secure connection with the Back-end database.

In RFID systems, a RFID tag transmits information to a nearby reader using RF signals. The RF communication used in RFID systems makes it vulnerable to various attacks such as eavesdropping, traffic analysis, message interception and impersonation (e.g., spoofing and replay). Among the various attacks, the impersonation attack permits an adversary to fool RFID systems. For example, (1) In case of a spoofing attack, an adversary can replace a tag of an expensive item with a bogus tag which transmits data obtained from a cheaper item in response to a query from a nearby reader. The tag of an expensive item is attached to some one in shop. When the expensive item passes the checkout counter, a price of the cheaper item is charged for the expensive item and the expensive item is still perceived as existent one in shop. (2) In case of a replay attack, an adversary can impersonate the tag by retransmitting previously transmitted message between a tag and a reader. Therefore, these attacks allow an adversary to fool RFID systems. To prevent these attacks, RFID systems should provide mutual authentication between a reader and a tag to assure that no adversaries can make valid message.

In this paper, we study mutual authentication scheme as an efficient method to resolve these problems, especially for low-cost systems. A low-cost RFID tag is limited in computing power, communication mechanism and storage space since a RFID chip with approximately 4,000 gates is considered to be low-cost. This implies that previously classical authentication schemes are not suitable. Therefore, it is important to construct an efficient authentication scheme for low-cost RFID systems. Furthermore, we will face up to ubiquitous computing environment in the near future. It is also important to construct protocol which is well suitable to ubiquitous computing environment. In this paper, we propose mutual authentication protocol which is suitable to ubiquitous computing environment.

## 1.1 Related Work

Researchers have recognized the privacy problem of RFID tags [8] and are continuing to devise better approaches to protect a user privacy. We describe some of the related studies below. The simplest physical approach for the protection of user privacy is to “kill” RFID tags [10] before they was put in the hands of a user. However, a low-cost RFID tag will be used in numerous applications and many of these applications may require that tags maintain active state in the hands of a user. Therefore, this method is not a useful solution. In addition to “kill” method, other physical methods are Faraday Cage and active jamming [3]. In addition to “kill” method, other physical methods are Faraday Cage and active jamming [3]. However, two methods are also not suitable to protect a user privacy.

Another general approach is using encryption algorithm. In this approach, messages are encrypted using asymmetric public key algorithms [1, 2, 4] which

are based on re-encryption method. In [2], Juels *et al.* proposed a scheme to protect a user privacy implications of RFID-tags embedded in banknotes. The resulting ciphertext undergoes periodic re-encryption period. Recently, Avoine described the privacy issues in RFID banknote protection scheme [1]. In [1, 4], these schemes are based on universal re-encryption used in Mixnets. However, this approach cannot protect a user privacy from a malicious reader. If a malicious reader only receives a response from a tag and do not perform re-encryption operation, then the malicious reader can obtain constant ciphertext. Using this process consecutively, the malicious reader obtains user's location history.

The other approach is to design an authentication protocol using one-way hash function [5, 6, 10]. This approach can prevent an exposure of tag ID using one-wayness property of hash function. However, schemes of papers [5, 6, 10] provide partial solutions to protect a user privacy. In [10], whenever a tag receives a query from a reader, the tag responds with its *metaID* which is fixed. Therefore, an adversary can trace the tag using *metaID*. Ohkubo *et al.* proposed a protocol using a hash chain mechanism [6]. This method uses two different hash functions to protect a user privacy. However, the Back-end database should compute all the hash chains, i.e., it is impractical. However, an adversary can attack these schemes [6] using eavesdropping or impersonation attack. Henrici *et al.* also proposed a simple scheme [5], called hash-based ID variation scheme (HIDV), using one-way hash function and the scheme enhances location privacy by changing traceable identifiers on every session. The proposed scheme is not secure against impersonation attack such as spoofing. Recently, LEE *et al.* [9] proposed LCAP protocol which improved HIDV scheme in both efficiency and security. Also, Rhee *et al.* proposed challenge-response based RFID authentication protocol (CRAP) which is suitable to ubiquitous computing environment [7].

## 1.2 Contribution

We propose an efficient authentication protocol, OHLCAP, for Hash-based low-cost RFID systems, which is suitable to ubiquitous computing environment. In Table 1, we show efficiency analysis with respect to computation cost and security against various threats in LCAP, CRAP, and OHLCAP. Also, we consider whether the schemes are suitable to ubiquitous computing environment or not.

- *Application:* In ubiquitous computing environment, components of the RFID systems can exist in anywhere. As schemes described in papers [5, 9], if a tag's ID should be dynamic value to protect a user privacy, the tag only communicates with a fixed Back-end database since the tag must synchronize the tag's dynamic ID value with the Back-end database. However if a tag's ID is static value such as CRAP [7], then the tag can perform authentication protocol with any Back-end database since the scheme does not need synchronization of the tag's ID between a Back-end database and the tag. Therefore, the tag holding static ID is able to communicate with any reader in ubiquitous computing environment. As shown in Table 1, OHLCAP is suitable to ubiquitous computing environment because of using static ID. Although our protocol uses static ID, it is secure against various attacks.

**Table 1.** The analysis of efficiency and security

Protocol		LCAP	CRAP	OHLCAP
<i>Memory.</i>	Tag	$1l$	$1l$	$5l$
	Back-end database	$6l$	$1l$	$4l$
<i>Computation.</i>	Tag	$2H$	$3H$	$1H (+A)$
	Back-end database	$1H$	$(\frac{N}{2} + 1)H$	$1H + \varepsilon$
<i>Communication.</i>	Tag $\rightarrow$ Reader	$1\frac{1}{2}l$	$2l$	$2\frac{1}{2}l$
	Reader $\rightarrow$ Tag	$\frac{1}{2}l$	$l$	$\frac{1}{2}l$
Spoofing		Prevention	Prevention	Prevention
Loss of message		Restoration	–	–
Replay attack		Prevention	Prevention	Prevention
Location privacy		Prevention	Prevention	Prevention
Distributed database environment		Unsuitability	Suitability	Suitability

**Notations of Table:**  $l$  : the output size of a one-way hash function or the length of ID,  $H$  : the cost of a one-way hash function operation,  $N$  : the number of tags in a Back-end database,  $A$  : the cost of additional operations except for hash operation in a tag,  $\varepsilon$  : the cost of additional operations except for hash operation in a Back-end database, – : No consideration.

- *Memory.* : the storage cost of each entity.
- *Computation.* : the maximum computation cost of each entity during the execution of an authentication protocol.
- *Communication.* : the length of bits that a tag and a reader send during the execution of an authentication protocol.

- *Efficiency* : As shown in Table 1, we consider a storage cost, a communication cost, and a computation cost of each entity. As compared with the previously proposed schemes in Table.1, although OHLCAP stores more secret values than both LCAP and CRAP, OHLCAP requires that a tag only operates *one* one-way hash function operation, and additional operations  $A$  which are four xor-operations and one addition operation. Since both xor-operation and addition operation are very simple bits operation, hardware embodiment of these operations is simpler than one-way Hash function. Therefore, OHLCAP is suitable to a low-cost RFID tag.

ORGANIZATION OF THE PAPER. This paper is organized as follows: In Section 2, we describe security and privacy risks in RFID systems. We describe our scheme OHLCAP in Section 3. In Section 4, we analyze our scheme in security. Finally, we conclude in Section 5.

## 2 Security and Privacy Risks

### 2.1 Security Risks

In RFID systems, since an adversary can monitor all messages transmitted in wireless communication between a reader and a tag, the adversary can infringe upon

a person's privacy using various methods. Therefore, RFID systems must be designed to be secure against attacks such as eavesdropping, traffic analysis, message interception and impersonation (e.g., spoofing and replay) as described below.

**Passive attack - Eavesdropping:** A passive adversary can eavesdrop on messages between a reader and a tag. By eavesdropping, the adversary may obtain a user's secret information. So, RFID systems should be designed that the eavesdropper cannot get any secret information from the eavesdropped messages.

**Active attack - Impersonation:** An active adversary can query to a tag and a reader in RFID systems. By this property, the adversary can impersonate the target tag or reader. There are two types of impersonation attack; replay and spoofing. Besides of impersonation attack, an active adversary can try to trace the location of a tag using traffic analysis : distinguishing whether the response is transmitted by the target tag or not. Therefore, RFID systems should be designed that an active adversary cannot impersonate a target tag or reader and distinguish a target tag's response from a random value.

**Active attack - Message interception:** In this attack, although an adversary cannot obtain any information in RFID systems, message interception makes a target tag unable to operate further. Among the previously proposed schemes, several schemes such as [5, 9] require that a tag should receive some value from a Back-end database and update stored values using the received value. If message interception occurs in these schemes, the Back-end database should be able to restore the messages. In result, RFID systems can normally operate. Therefore, RFID systems should be able to detect message interception except that a tag does not need to receive updating values from a reader for next session.

## 2.2 Privacy Risks

As mentioned above, an adversary is able to attack RFID systems using various methods. These attacks make a tag able to disclose sensitive information to an unauthorized reader. If a link between a tag and a user holding the tag is established, his movement can be traced by tracking the tag's ID. This implies that the adversary infringes a user's location privacy. To design secure RFID systems, we should consider these risks in detail below.

– *Information Leakage* : A person is prone to carrying various tagged objects in every life. Some of objects such as expensive products and medicine are quite personal and provide information that the user does not want anyone to know. In RFID systems, the tag emits only distinguishable information in response to a query from a nearby reader. So, various personal information can be leaked without the acknowledgement of the user.

– *Traceability* : When a target tag transmits a response to a nearby reader, an adversary can record the transmitted message and can establish a link between the response and the target tag. Once a link established, the adversary is able to know the user's location history.

### 3 Our Hash-Based Low-Cost Authentication Protocol

In this section, we describe our OHLCAP for Hash-based low-Cost RFID systems. OHLCAP consists of set-up and mutual authentication phases.

#### 3.1 System Set-Up

Let  $H : \{0, 1\}^* \rightarrow \{0, 1\}^l$  be a one-way hash function, where a hash value space belongs to  $\{0, 1\}^l$ . ID denotes identity of a tag and is a unique value in  $\{0, 1\}^l$ . In the set-up phase, both a tag and a Back-end database store several secret values and tag's ID. Data fields of a tag and a reader are initialized to the following values:

1. Back-end Database : First, a Back-end database divides identities of tags into several groups. If a number of system's tags are  $N(= mn)$ , a Back-end database divides it into  $n$  groups which include  $m$  identities of tags and generates a group index GI in each group, as shown Figure 1. Then, data fields of a Back-end database are initialized to GI, ID, K, S and DATA. The Back-end database needs a one-way hash function to execute hash function operation.
  - GI is a group index of tags with  $l$ -bit string. If a tag belongs to  $i$ -th group,  $GI_i$  is a group index of the tag.
  - K is a secret value with  $l$ -bit string and is stored in all tags. S is a tag's secret value with  $l$ -bit string.
  - ID is  $l$ -bit string, which is used for identifying. Tag's IDs differs from group indices  $GI_i, i \in \{1, \dots, n\}$ .
  - DATA stores an accessible information about each tag, e.g., a secret value S.
2. Reader : A reader picks uniformly a random value  $r$  with  $\{0, 1\}^l$ . A reader does not need to execute any operation. A reader merely forwards a tag's message (or a Back-end database's message) to a Back-end database (or a tag).
3. Tag : The data field of a tag is initialized to its own ID, GI, K and S,  $c$ . The tag stores ID, GI, K, S, and a counter  $c$ . The counter  $c$  is initialized

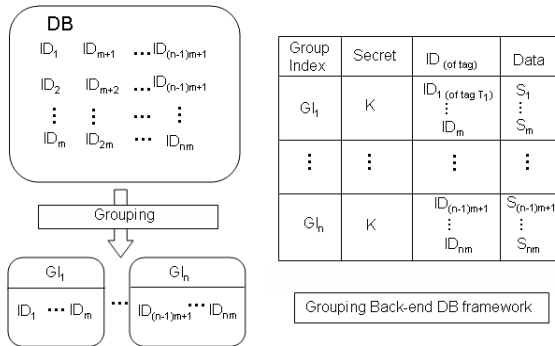


Fig. 1. Back-end database framework

by an arbitrary value, which is  $l$ -bit string. Whenever a tag receives a query from a nearby reader, the tag increase a counter  $c$ . To execute a one-way hash function operation, the tag needs a one-way hash function.

### 3.2 Mutual Authentication - OHLCAP

When a reader queries to a tag, the tag and the reader authenticate each other as shown in Figure 2. To help to understand OHLCAP protocol, we assume that the tag belongs to  $i$ -th group.

NOTATIONS. The addition operation of bits is denoted by  $+$  and the exclusive-or (xor) operation of bits is denoted by  $\oplus$ .  $m||w$  denotes the concatenation of two messages,  $m$  and  $w$ .

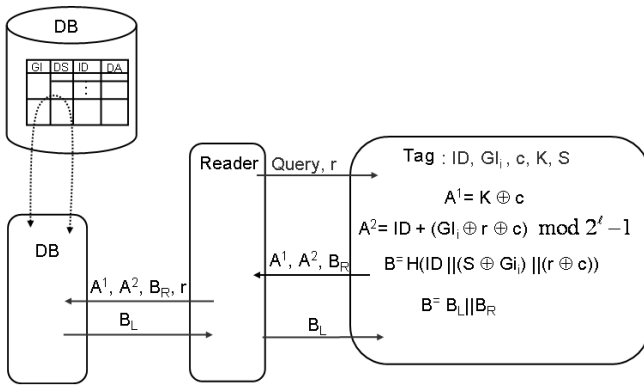


Fig. 2. OHLCAP protocol

**Step 1.** A reader picks a random value  $r$  and sends **Query** and  $r$  to a nearby tag.

**Step 2.** To respond to the query of the reader, the tag checks a random value  $r$  whether it is all zero value or not.

1. If  $r$  value is all zero, the tag sends “stop” message to the reader and halts the protocol.
2. Otherwise, the tag performs processes as follows.
  - The tag computes  $A^1 = K \oplus c$ ,  $A^2 = ID + (GI_i \oplus r \oplus c) \bmod (2^l - 1)$  using  $r$ ,  $c$  and its own  $ID$ ,  $GI_i$  and  $K$ .
  - Also, the tag computes  $B = H(ID || (S \oplus GI_i) || (r \oplus c))$  using  $ID$ ,  $c$ ,  $r$ ,  $GI_i$  and  $S$ , and sends  $A^1$ ,  $A^2$  and  $B_R$  to the reader, where  $B_R$  is a right half of  $B$ , so  $B_R$  has the length of  $\frac{1}{2}l$  bit.
  - Then, the tag increases the counter  $c$  which should not exceed  $2^l - 1$ . If the counter  $c$  exceeds  $2^l - 1$ , it is initialized by initial  $c$ .

- Step 3.** Upon receiving  $A^1$ ,  $A^2$  and  $B_R$  from the tag,
1. The reader forwards  $A^1$ ,  $A^2$ ,  $B_R$  and  $r$  to the Back-end database.
  2. The Back-end database computes  $c' = A^1 \oplus K$  and  $ID'_j = A^2 - (GI_j \oplus r \oplus c') \bmod (2^l - 1)$  using all group indices  $GI_j$ ,  $j \in \{1, \dots, n\}$ .
  3. The Back-end database checks if one of computed  $ID'_{j \in \{1, \dots, n\}}$  is matching to one of the stored IDs in the Back-end database. If this process succeeds, the Back-end database check if the  $GI_j$  used to compute  $ID'_j$  is equal to the group index  $GI_i$  that contains the matching  $ID'_j$ .
    - If this succeeds, the Back-end database computes  $H(ID || (S \oplus GI_i) || (r \oplus c))$  using  $c$ ,  $r$ ,  $GI_i$ ,  $S$  and the matched ID.
    - Otherwise, the Back-end database halts this process.
  4. Then, the Back-end database authenticates the tag by checking if the right half of the computing value  $H(ID || (S \oplus GI_i) || (r \oplus c))$  is equal to the received value  $B_R$ .
  5. The Back-end database sends  $B_L$  to the reader, where  $B_L$  is a left half of  $B$ . the reader forwards  $B_L$  to the tag.
- Step 4.** The tag authenticates the reader by checking if the received value  $B_L$  is equal to the left half of  $B$  of step 2.

## 4 Security Analysis

In this section, we analyze our protocol OHLCAP in security. Considering attack methods in described section 2.2, we analyze the security of our protocol against the threats introduced in section 2.2; *information leakage* and *traceability*.

**Information Leakage.** In OHLCAP, an adversary must be authenticated to get any sensitive information in a tag. To pass authentication protocol without knowing  $GI$ ,  $K$ ,  $c$ ,  $S$  and  $ID$ , an adversary only must guess  $B_L$  value after collecting messages  $A^1$ ,  $A^2$ , and  $B_R$ . However, because of one-wayness property of hash function  $H$ , the adversary cannot get sensitive information of  $B_L$  from  $A^1$ ,  $A^2$  and  $B_R$ . In OHLCAP, since an adversary does not know a secret  $K$ , even if the adversary eavesdrops  $A^1$ , the adversary cannot get the tag's group index  $GI$ . So, the adversary cannot get any information of  $B_L$  from  $A^2$ . Therefore, the adversary has to randomly pick a string from  $\{0, 1\}^{\frac{1}{2}l}$ . Also, even if an adversary collects the hash values  $B_L$ ,  $B_R$ , the adversary cannot get information of tag's ID. In order to guess the target tag's ID, the adversary has to randomly select a string from  $\{0, 1\}^l$  by one-wayness property of hash function  $H$ . Therefore, the advantage of the adversary is at most  $\frac{1}{2^{(l/2)}} + \frac{1}{2^l}$ , which is negligible.

**Traceability.** Our OHLCAP protocol guarantees location privacy by using refreshed values  $r, c$ , where  $r$  and  $c$  are refreshed by a reader and a tag in each session, respectively. Even if a malicious reader does not refresh a random value  $r$ , a tag transmits the refreshed values that are refreshed by a counter  $c$ , where the counter  $c$  is refreshed by a tag in each session.

– In OHLCAP protocol, an adversary can eavesdrop on  $A^1$ ,  $A^2$  in between a reader and a target tag. Since the adversary does not know secret  $K$ , she is not



able to extract the  $c$  value. Thus, the adversary cannot obtain the tag's group index  $GI$  from eavesdropped messages  $A^1, A^2$ . Therefore, it is impossible that the adversary obtains the target tag's ID. This means that the adversary cannot trace the target tag.

– In OHLCAP protocol, since all tags in one group uses an identical group index  $GI$  and a secret  $K$ , we consider a special attack that an adversary obtains secret value  $K$  and some  $GI_j$  ( $j \in \{1, \dots, n\}$ ) by only attacking physically some tag. This tag is unable to operate further. The adversary try to attack OHLCAP using obtained values. First, the adversary eavesdrops on  $A^1, A^2, B_L$  and  $B_R$  between a reader and a target tag. Then, the adversary can extract a counter  $c$  from  $A^1$  using the value  $K$  and compute a some  $ID'$  from  $A^2$  by using obtained values  $GI_j$ . The adversary does not know whether a computed  $ID'$  is the tag's ID or not. So, by using one-way hash function, the adversary should check if eavesdropped value  $B$  is equal to  $H(ID' || (S \oplus GI_j) || (r \oplus c))$ . However, the adversary does not know a secret value  $S$ . Therefore, the adversary is not able to check if a computed  $ID'$  is the target tag's ID, and cannot compute the target tag's ID. Thus, the adversary cannot trace the target tag.

In RFID systems, as mentioned in section 2.1, an adversary can attack various attacks such as eavesdropping, traffic analysis, message interception and impersonation. In order to analyze about a user privacy protection of our protocol, we only consider attacks such as eavesdropping and traffic analysis. Now, we show that our protocol is secure against remaining attacks such as message interception and impersonation(e.g., spoofing and replay).

**Impersonation.** In our protocol, impersonation attack can be prevented by mutual authentication between a reader and a tag. In OHLCAP, an adversary cannot impersonate a target tag using *a replay attack* since the valid message is refreshed in each session by a random value  $r$  and a counter  $c$ . Also, an adversary queries a target tag by impersonating as a reader, receives messages back from the target tag, then she may try a spoofing attack to impersonate the target tag. However, without knowing  $ID, GI, S$  of the target tag, the adversary is unable to compute a half right  $B_R$  of the  $B$  that can only be generated by the target tag. Therefore, it is impossible to impersonate the target tag by *a spoofing attack* in OHLCAP.

In OHLCAP, a tag does not receive any message from a reader in order to update own ID. Even if loss of message occurs between a tag and a reader, the tag increases a counter  $c$  by itself and computes  $A^1, A^2, B_R$  using  $r$  received from a nearby reader in next session. Therefore, message intercetpion does not need to be considered in OHLCAP.

## 5 Conclusion

We have proposed an efficient and secure authentication protocol OHLCAP to protect a user privacy, especially for low-cost RFID systems in ubiquitous computing environment. The proposed scheme needs only *one* one-way hash function

operation and hence is quite efficient. Leakage of information is prevented in the scheme since a tag emits its information only after authentication. By refreshing a message transmitted from a tag in each session, OHLCAP also provides a location privacy and is secure against many attacks such as eavesdropping, traffic analysis, message interception, spoofing and replay.

## References

1. G. Avoine. *Privacy issues in RFID banknote protection schemes*. In international Conference on Smart Card Research and Advanced Applications - CARDIS, Toulouse, pp.22-27, 2004.
2. A. Juels and R. Pappu. *Squealing euros : Privacy protection in RFID-enabled banknotes*. In proceedings of Financial Cryptography -FC'03, vol.2742 LNCS, pp.103-121, Springer-Verlag, 2003.
3. A. Juels, R. L. Rivest and M. Szudlo. *The Blocker Tag: Selective Blocking of RFID tags for Consumer Privacy*. In the 8th ACM Conference on Computer and Communications Security, pp. 103-111, ACM Press, 2003.
4. S. Junichiro, R. Jae-Cheol and S. Kouichi, *Enhancing privacy of Universal Re-encryption scheme for RFID Tags*. EUC 2004, Vol. 3207 LNCS, pp.879-890, Springer-Verlag, 12, 2004
5. D. Henrici and P. Muller. *Hash-based Enhancement of Location Privacy for Radio-Frequency Identification Devices using Varying Identifiers*. PerSec'04 at IEEE Per-Com. 2004
6. M. Ohkubo, K. Suxuki and S. Kinoshita. *Efficient Hash-Chain Based RFID Privacy Protection Scheme*. Ubcomp2004 workshop.
7. Keunwoo Rhee, Jin Kwak, Seungjoo Kim and Dongho Won. *Challenge-Response Based RFID Authentication Protocol for Distributed Database Environmnet.*, SPC 2005, LNCS 3450, pp. 70-84, 2005.
8. S. E. Sarma, S. A. Weis and D. W. Engels. *Radio-frequency identification systems*. CHES'02, vol.2523 LNCS, pp.454-469, Springer-Verlag, 2002.
9. L. Su Mi, H. Young Ju, L. Dong Hoon and L. Jong In. *Efficient Authentication for Low-Cost RFID systems*. ICCSA05, vol. 3480 LNCS, pp.619-629, Springer-Verlag, 2005.
10. S. A. Weis, S. E. Sarma, S. A. Weis and D. W. Engels. *Security and privacy Aspects of Low-Cost Radio Frequency Identification Systems*. First International Conference on Security in Pervasive Computing, 2003. <http://theory.lcs.mit.edu/sweis/spc-rfid.pdf>

# A New Simple Authenticated Key Agreement and Protected Password Change Protocol

Eun-Jun Yoon and Kee-Young Yoo\*

Department of Computer Engineering, Kyungpook National University,  
Daegu 702-701, South Korea  
Tel.: +82-53-950-5553; Fax: +82-53-957-4846  
ejyoon@infosec.knu.ac.kr, yook@knu.ac.kr

**Abstract.** In 2005, Chang et al. proposed a simple authenticated key agreement and protected password change protocol. However, Chang et al.'s schemes are still susceptible to stolen-verifier attack and Denial-of-Service attacks. Accordingly, the current paper demonstrates the vulnerability of Chang et al.'s schemes to two simple attacks and then presents an improved scheme to resolve such problems. In contrast to Chang et al.'s protected password change protocol, the proposed protected password change protocol can securely update user passwords without a complicated process, while also providing greater security.

**Keywords:** Cryptography, Authentication, Password, Key agreement, Password change, Dffie-Hellman key agreement.

## 1 Introduction

The Dffie-Hellman key agreement scheme [1], which developed to provide a common session key between two parties, is an epochal breakthrough that can produce a common session key without any prior common information. However, this method has a weakness of possible man-in-the middle attacks [2]. Recently, Seo and Sweeney [3] proposed a new key agreement protocol based on the Diffie-Hellman protocol called the simple authenticated key agreement algorithm (SAKA). In the SAKA protocol, two parties have a pre-shared password [4] for data communication, produce a session key by exchanging messages, and confirm each other. Because they can simplify key agreement, SAKA-like protocols are widely used in research on key agreement, and therefore there have been numerous attempts to enhance SAKA-like protocols.

In 2002, Yeh and Sun [5], and Kobara and Imai [6] combined the pre-shared password technique and the Diffie-Hellman scheme to achieve the same purpose as the SAKA-like schemes. In 2005, Chang et al. [7] modified Yeh and Sun's SAKA-like protocol to improve its efficiency and also proposed a protected password change protocol to achieve user authentication and to arbitrarily change a password. Chang et al.'s key agreement protocol requires fewer steps and less computation cost than the Kobara-Imai scheme. Moreover, Chang et al. not

---

\* Corresponding author.

only give a heuristic security analysis, but also formally prove it using Ballare, Poincheval and Rogaway's model [8].

However, Chang et al.'s schemes are still susceptible to stolen-verifier attack [9], in which obtaining the secret data stored in a server can allow an illegitimate user to login to a server as a legitimate user. Additionally, their protected password change protocol suffers from a Denial-of-Service attacks [9], in which an attacker can easily make the server reject all subsequent login requests from any user. Accordingly, the current paper demonstrates that Chang et al.'s schemes are vulnerable to stolen-verifier attack and Denial-of-Service attacks and also presents an improved scheme to isolate such problems. In contrast to Chang et al.'s protected password change protocol, the proposed schemes can securely update user passwords without a complicated process, while also providing greater security.

The remainder of this paper is organized as follows: Section 2 defines the security properties. Section 3 briefly reviews Chang et al.'s schemes, then Section 4 demonstrates stolen-verifier attack and Denial-of-Service attacks with their schemes. The proposed schemes are presented in Section 5, while Section 6 discusses the security of the proposed schemes. Our conclusions are presented in Section 7.

## 2 Security Properties

The following security properties of the authentication protocols should be considered. Password authentication protocols are very subject to replay, password guessing, and stolen-verifier attacks [9].

- (1) **Replay attack:** A replay attack is an offensive action in which an adversary impersonates or deceives another legitimate participant through the reuse of information obtained in a protocol.
- (2) **Guessing attack:** A guessing attack involves an adversary simply (randomly or systematically) trying passwords, one at a time, in hope that the correct password is found. Ensuring that passwords are chosen from a sufficiently large space can resist exhaustive password searches. However, most users select passwords from a small subset of the full password space. Such weak passwords with low entropy are easily guessed by using so-called dictionary attack.
- (3) **Stolen-verifier attack:** In most applications, the server stores verifiers of users' passwords (e.g., hashed passwords) instead of the clear text of passwords. The stolen-verifier attack means that an adversary who steals a password-verifier from the server can use it *directly* to impersonate a legitimate user in a user authentication execution. Note that the main purpose of an authentication scheme against the stolen-verifier attack is to reduce the immediate danger to user authentication. In fact, an adversary who has a password-verifier may further mount a guessing attack.

Password change protocols allow an authenticated user to change his/her password. Besides those attacks mentioned above, a password change protocol is very vulnerable to Denial-of-Service attacks [9].

- (1) Denial-of-Service attack: A Denial-of-Service attack prevents or inhibits the normal use or management of communications facilities. This attack may be directed to a specific user. For example, an adversary may perform this attack to cause the server to reject the login of a specific user.

In addition, the following security properties of session key agreement protocols should be considered since they are often desirable in some environments [10].

- (1) Implicit key authentication: Implicit key authentication is the property obtained when identifying a party based on a shared session key, which assures that no other entity than the specifically identified entity can gain access to the session key.
- (2) Explicit key authentication: Explicit key authentication is the property obtained when both implicit key authentication and key confirmation hold.
- (3) Mutual authentication: Mutual authentication means that both the client and server are authenticated to each other within the same protocol, while explicit key authentication is the property obtained when both implicit key authentication and key confirmation hold.
- (4) Perfect forward secrecy: Perfect forward secrecy means that if a long-term private key (e.g. user password or server private key) is compromised, this does not compromise any earlier session keys. In password authentication with key distribution, forward secrecy is a highly desirable security feature.

### 3 Review of Chang et al.'s Schemes

This section briefly reviews Chang et al.'s key agreement protocol and protected password change protocol and then show how stolen-verifier attacks and Denial-of-Service attacks can work on their protocol. Abbreviations used in this paper are as follows:

- $id$ : public user identity of client.
- $pw$ : secret and possibly weak user password.
- $K$ : strong secret key of server.
- $p, q$ : large prime numbers  $p$  and  $q$  such that  $q|p-1$ .
- $g$ : generator with order  $q$  in the Galois field  $GF(p)$ , in which Diffie-Hellman problem is considered hard.
- $a, b$ : session-independent random exponents  $\in [1, q-1]$  chosen by client and server, respectively.
- $sk$ : shared session key computed by client and server.
- $H(\cdot)$ : strong one-way hash function.
- $\oplus$ : bit-wise XOR operation.

#### 3.1 Chang et al.'s Simple Authenticated Key Agreement Protocol

Chang et al.'s simple authenticated key agreement protocol works as follows:

Step 1. Client  $\rightarrow$  Server:  $R_A \oplus pw$

The client chooses a random number  $a \in [1, q - 1]$ , computes  $R_A = g^a \bmod p$ , and then sends  $R_A \oplus pw$  to the server.

Step 2. Server  $\rightarrow$  Client:  $R_B || H(K_B, R_A)$

After receiving  $R_A \oplus pw$ , the server recovers  $R_A$  by computing  $(R_A \oplus pw) \oplus pw$ . Then the server chooses a random number  $b \in [1, q - 1]$ , computes  $R_B = g^b \bmod p$  and  $K_B = R_A^b = g^{ab} \bmod p$ , and then sends  $R_B || H(K_B, R_A)$  to the client.

Step 3. Client  $\rightarrow$  Server:  $H(K_A, R_B)$

After receiving  $R_B || H(K_B, R_A)$ , the client computes  $K_A = R_B^a = g^{ab} \bmod p$  and verifies whether the received  $H(K_B, R_A)$  is equal to  $H(K_A, R_A)$ . If it holds, the client computes  $H(K_A, R_B)$  and sends it to the server.

Step 4. Server  $\rightarrow$  Client: *Access granted / denied*

After receiving  $H(K_A, R_B)$ , the server verifies whether  $H(K_A, R_B)$  is equal to  $H(K_B, R_B)$ . If it is equal, the client and server agree on the common session key  $Key = H(K_A) = H(K_B) = H(g^{ab} \bmod p)$ .

### 3.2 Chang et al.'s Protected Password Change Protocol

In Chang et al.'s protected password changing protocol, the server allows the client to change their old password  $pw$  to a new password  $newpw$ . Chang et al.'s protected password change protocol works as follows:

Step 1\*. Client  $\rightarrow$  Server:  $R_A \oplus pw || R_A \oplus newpw$

The client chooses a random number  $a \in [1, q - 1]$ , computes  $R_A = g^a \bmod p$ , and then sends  $R_A \oplus pw || R_A \oplus newpw$  to the server.

Step 2\*. Server  $\rightarrow$  Client:  $R_B || H(K_B, R_A)$

After receiving  $R_A \oplus pw || R_A \oplus newpw$ , the server recovers  $R_A$  by computing  $(R_A \oplus pw) \oplus pw$  and uses the recovered  $R_A$  to derive  $newpw$  by computing  $(R_A \oplus newpw) \oplus R_A$ . Then the server chooses a random number  $b \in [1, q - 1]$ , computes  $R_B = g^b \bmod p$  and  $K_B = R_A^b = g^{ab} \bmod p$ , and sends  $R_B || H(K_B, R_A)$  to the client.

Step 3\*. Client  $\rightarrow$  Server:  $H(K_A, R_B) \oplus newpw$

After receiving  $R_B || H(K_B, R_A)$ , the client computes  $K_A = R_B^a = g^{ab} \bmod p$  and verifies whether the received  $H(K_B, R_A)$  is equal to  $H(K_A, R_A)$ . If it holds, the client computes  $H(K_A, R_B) \oplus newpw$  and sends it to the server.

Step 4\*. Server  $\rightarrow$  Client: *Access granted / denied*

After receiving  $H(K_A, R_B) \oplus newpw$ , the server uses the recovered  $newpw$  in Step 2\* to derive  $H(K_A, R_B)$  by computing  $(H(K_A, R_B) \oplus newpw) \oplus newpw$ . Then the server verifies whether the recovered  $H(K_A, R_B)$  is equal to  $H(K_B, R_B)$  or not. If it is equal, the client and server have successfully changed their shared password  $pw$  to the new password  $newpw$  and they can agree on the common session key  $Key = H(K_A) = H(K_B) = H(g^{ab} \bmod p)$ .

## 4 Cryptanalysis of Chang et al.'s Schemes

This section shows that Chang et al.'s schemes is vulnerable to a Denial-of-Service attacks and a stolen-verifier attack.

### 4.1 Denial-of-Service Attack on Chang et al.'s Protected Password Change Protocol

Usually, the server closes a login session if the number of error login attempts of an account exceeds a limited value (e.g. 3 times). Even so, such a client's account is still workable and later login requests will pass as long as the correct password is provided. However, Chang et al.'s protected password change protocol can suffer from a Denial-of-Service attacks, in which an attacker can easily make the server reject all subsequent login requests from any client.

In Step 1\* of Chang et al.'s protected password change protocol, an attacker can simply replace new password digest  $R_A \oplus newpw$  with forged new password digest  $R_A \oplus newpw \oplus X$ , where  $X$  is a random number chosen by the attacker. After receiving the replaced messages  $(R_A \oplus pw || R_A \oplus newpw \oplus X)$ , the server can retrieve  $R_A$  from  $R_A \oplus pw$  by computing  $R_A \oplus pw \oplus pw$ . Then, the server uses the recovered  $R_A$  to obtain modified new password  $newpw \oplus X$  from  $R_A \oplus newpw \oplus X$  by computing  $R_A \oplus newpw \oplus X \oplus R_A$ .

In step 3\*, an attacker can replace  $H(K_A, R_B) \oplus newpw$  with  $H(K_A, R_B) \oplus newpw \oplus X$  by using the chosen random number in Step 1\*. After receiving the replaced messages  $H(K_A, R_B) \oplus newpw \oplus X$ , the server can retrieve  $H(K_A, R_B)$  from  $H(K_A, R_B) \oplus newpw \oplus X$  using the obtained value  $newpw \oplus X$  in the Step 1\* and checks whether  $H(K_A, R_B)$  is equal to  $H(K_B, R_B)$  holds or not. Because it holds, the server will pass the authentication and update a new password as  $newpw \oplus X$ . If value  $newpw \oplus X$  is not equal to the client's new password  $newpw$ , all subsequent login requests of that client will be rejected until that client has re-registered with the server. Therefore, Chang et al.'s protected password change protocol is insecure against Denial-of-Service attacks.

### 4.2 Stolen-Verifier Attack on Chang et al.'s Schemes

Servers are always the target of attacker, because numerous clients' secrets are stored in their databases. The client password stored in the server can be eavesdropped and then used to impersonate as the original client. Chang et al. do not explain stolen-verifier attacks, where obtaining the client password  $pw$  stored in a server can allow an illegitimate client to login to the server as a legitimate client.

In Chang et al.'s protected password change protocol (which includes Chang et al.'s simple authenticated key agreement protocol), if an attacker has stolen the password  $pw$  from the server, he or she can choose a random number  $a'$ , computes  $R'_A = g^{a'} \bmod p$ , choose a new password  $newpw'$ , and uses  $R'_A$  to compute client password digest  $R'_A \oplus pw$  and client new password digest  $R'_A \oplus newpw'$  in Step 1\*. Then the attacker can send its as a login request to the server and can impersonate the original client. Therefore, Chang et al.'s schemes is insecure against stolen-verifier attacks.

## 5 Proposed Schemes

This section proposes an improved simple authenticated key agreement protocol and protected password change protocol to overcome the above mentioned problems inherent in Chang et al.'s scheme. In the proposed scheme, the server stores  $vpw = (H(id, pw) \oplus K) + K$  using the server's secret key  $K$  instead of  $pw$  for each client in the database to overcome the stolen-verifier attack.

### 5.1 Proposed Simple Authenticated Key Agreement Protocol

The proposed simple authenticated key agreement protocol works as follows:

- Step 1 Client  $\rightarrow$  Server:  $id || R_A \oplus H(id, pw)$   
 The user submits his  $id$  and  $pw$  to the client. The client then chooses a random number  $a \in [1, q - 1]$ , computes  $R_A = g^a \text{ mod } p$ , and then sends  $id || R_A \oplus H(id, pw)$  as a login request to the server.
- Step 2 Server  $\rightarrow$  Client:  $R_B || H(K_B, R_A)$   
 After receiving  $id || R_A \oplus H(id, pw)$ , the server retrieves  $R_A$  from  $R_A \oplus H(id, pw)$  by computing  $R_A \oplus H(id, pw) \oplus (vpw - K) \oplus K$ . Then, the server chooses a random number  $b \in [1, q - 1]$  and computes  $R_B = g^b \text{ mod } p$  and  $K_B = (R_A)^b = g^{ab} \text{ mod } p$ . Then, the server uses its own  $K_B$  and the recovered  $R_A$  to compute  $H(K_B, R_A)$ . The server then sends  $R_B || H(K_B, R_A)$  as the server's authentication token to the client.
- Step 3 Client  $\rightarrow$  Server:  $id || H(K_A, R_B)$   
 After receiving  $R_B || H(K_B, R_A)$ , the client computes  $K_A = (R_B)^a = g^{ab} \text{ mod } p$  and  $H(K_A, R_A)$ , and then verifies the consistency between the retrieved  $H(K_A, R_A)$  and the received  $H(K_B, R_A)$ . If the result is positive, the client computes  $H(K_A, R_B)$  and sends this client authentication token  $H(K_A, R_B)$  with the  $id$  to the server.
- Step 4 Server  $\rightarrow$  Client: *Access granted / denied*  
 After receiving  $id || H(K_A, R_B)$ , the server computes  $H(K_B, R_B)$  using its own copies of  $K_B$  and  $R_B$ , and then checks whether  $H(K_B, R_B) = H(K_A, R_B)$  holds or not. If it holds, the server can ensure the client is legal.

After mutual authentication between the client and the server,  $H(K_A) = H(K_B) = H(g^{ab} \text{ mod } p)$  is used as the session key, respectively.

### 5.2 Proposed Protected Password Change Protocol

The protected password change protocol allows a client to change his or her old password  $pw$  to a new password  $newpw$ . The proposed protected password change protocol works as follows:

- Step 1\* Client  $\rightarrow$  Server:  $id || R_A \oplus H(id, pw) || R_A \oplus H(id, newpw)$   
 The user submits his or her  $id$  and  $pw$  to the client. The client then chooses a random number  $a \in [1, q - 1]$ , computes  $R_A = g^a \text{ mod } p$



$p$ , chooses a new password  $newpw$ , and uses  $R_A$  to compute  $R_A \oplus H(id, pw)$  and  $R_A \oplus H(id, newpw)$ . Finally, the client sends  $id || R_A \oplus H(id, pw) || R_A \oplus H(id, newpw)$  as a login request to the server.

Step 2\* Server  $\rightarrow$  Client:  $R_B || H(K_B, R_A)$

After receiving  $id || R_A \oplus H(id, pw) || R_A \oplus H(id, newpw)$ , the server retrieves  $R_A$  from  $R_A \oplus H(id, pw)$  by computing  $R_A \oplus H(id, pw) \oplus (vpw - K) \oplus K$ . Then, the server uses the recovered  $R_A$  to obtain  $H(id, newpw)$  from  $R_A \oplus H(id, newpw)$  by computing  $R_A \oplus H(id, newpw) \oplus R_A$ . Then, the server chooses a random number  $b \in [1, q - 1]$  and computes  $R_B = g^b \text{ mod } p$  and  $K_B = (R_A)^b = g^{ab} \text{ mod } p$ . Then, the server uses its own  $K_B$  and the recovered  $R_A$  to compute  $H(K_B, R_A)$ . The server sends  $R_B || H(K_B, R_A)$  as the server's authentication token to the client.

Step 3\* Client  $\rightarrow$  Server:  $id || H(K_A, R_B, H(id, newpw))$

After receiving  $R_B || H(K_B, R_A)$ , the client computes  $K_A = (R_B)^a = g^{ab} \text{ mod } p$  and  $H(K_A, R_A)$ , then verifies the consistency between the retrieved  $H(K_A, R_A)$  and the received  $H(K_B, R_A)$ . If the result is positive, the client computes  $H(K_A, R_B, H(id, newpw))$  and sends this client authentication token  $H(K_A, R_B, H(id, newpw))$  with the  $id$  to the server.

Step 4\* Server  $\rightarrow$  Client: *Access granted / denied*

After receiving  $id || H(K_A, R_B, H(id, newpw))$ , the server computes the hash value  $H(K_B, R_B, H(id, newpw))$  using its own copies of  $K_B, R_B$  and the recovered  $H(id, newpw)$  in the Step 1\*, and then checks whether  $H(K_B, R_B, H(id, newpw)) = H(K_A, R_B, H(id, newpw))$  holds or not. If it holds, the server can ensure the client is legal and replaces  $vpw$  with  $(H(id, newpw) \oplus K) + K$ .

After mutual authentication between the client and the server,  $H(K_A) = H(K_B) = H(g^{ab} \text{ mod } p)$  is used as the session key, respectively.

## 6 Security Analysis

This subsection provides the security analysis of the proposed schemes. First, we define the security terms [10] needed for security analysis of the proposed schemes as follows:

**Definition 1.** A weak secret (password) is a value of low entropy  $W(k)$ , which can be guessed in polynomial time.

**Definition 2.** A strong secret key ( $K$ ) is a value of high entropy  $H(k)$ , which cannot be guessed in polynomial time.

**Definition 3.** The discrete logarithm problem (DLP) is explained by the following: Given a prime  $p$ , a generator  $g$  of  $Z_p^*$ , and an element  $\beta \in Z_p^*$ , find the integer  $\alpha$ ,  $0 \leq \alpha \leq p - 2$ , such that  $g^\alpha \equiv \beta \pmod{p}$ .

**Definition 4.** The Diffie-Hellman problem (DHP) is explained by the following: Given a prime  $p$ , a generator  $g$  of  $Z_p^*$ , and elements  $g^a \pmod{p}$  and  $g^b \pmod{p}$ , find  $g^{ab} \pmod{p}$ .

**Definition 5.** A secure one-way hash function  $y = H(x)$  is one where given  $x$ , computing  $y$  is easy and given  $y$ , computing  $x$  is hard.

Here, seven security properties: replay attack, password guessing attack, stolen-verifier attack, server spoofing attack, Denial-of-Service attack, mutual authentication, and perfect forward secrecy, must be considered for the proposed schemes. Under the above definitions, the following theorems are used to analyze seven security properties in the proposed schemes.

**Theorem 1.** *The proposed schemes can resist the replay attack.*

*Proof.* The attacker intercepts  $id||R_A \oplus H(id, pw)$  sent by the client in Step 1 and uses it to impersonate the client when sending the next login message. However, he/she has no ability to make a correct response  $id||H(K_A, R_B)$  in Step 3 because the random challenge  $R_A$  and  $R_B$  separately generated by the client and server are different every time. On the other hand, since the messages sent by the server and the client are different, the attacker cannot intercept any messages between them and then replay them to the other parties. Furthermore, obtaining  $R_A = g^a \bmod p$  and  $R_B = g^b \bmod p$  is computationally infeasible, as it is a discrete logarithm problem. Therefore, without knowing  $R_A$  and  $R_B$ , the attacker cannot impersonate the client or the server.

**Theorem 2.** *The proposed scheme can resist the password guessing attacks.*

*Proof.* On-line guessing attacks can be prevented by letting the server take appropriate intervals between trials. As described in Definition 1, weak passwords with low entropy are easily guessed in off-line guessing attacks. To avoid this problem, there must be no verifiable information on passwords in message exchanges. In the improved schemes, the password  $pw$  is protected by the client's random integer  $R_A$ . As such, no one can reveal the  $pw$  from the client's login message  $id||R_A \oplus H(id, pw)$  without knowing the client's random integer  $R_A$ . If the attacker wants to guess the client's password, he or she first must guess a password  $pw'$  and then finds  $R_A = R_A \oplus H(id, pw) \oplus H(id, pw')$ . However, the attacker has to break the discrete logarithm problem and Diffie-Hellman problem to find  $R_A = g^a \bmod p$  in Step 1 and  $K_B = g^{ab} \bmod p$  in Step 2, respectively. Hence, without knowing  $R_A$  and  $K_B$ , the attacker cannot verify the correctness of the guessed password by checking  $R_A \oplus H(id, pw) = R_A \oplus H(id, pw')$  in Step 1 and  $H(K_A, R_B) = H(K'_A, R_B)$  in Step 3, respectively. For the same reason, the attacker cannot guess session key  $H(K_A) = H(K_B) = H(g^{ab} \bmod p)$  from the server's response message  $R_B||H(K_B, R_A)$  in Step 2 or from the client's response message  $id||H(K_A, R_B)$  in Step 3 because  $H(\cdot)$  is a secure one-way hash function.

**Theorem 3.** *The proposed scheme can resist the stolen-verifier attack.*

*Proof.* Servers are always the target of attacks. An attacker may acquire  $vpw = (H(id, pw) \oplus K) + K$  stored in the server. However, without knowing the server's strong secret key  $K$ , the attacker cannot forge a login request to pass the authentication, as  $H(id, pw)$  is hidden in  $vpw = (H(id, pw) \oplus K) + K$  using the server's

strong secret key  $K$ . Thus, the correctness of the guessed password cannot be verified by checking  $(H(id, pw') \oplus K') + K' = vpw$ , where  $pw'$  is the guessed client's password and  $K'$  is the guessed server's strong secret key.

**Theorem 4.** *The proposed scheme can resist the server spoofing attack.*

*Proof.* The proposed schemes use the client's password  $H(id, pw)$  to ensure that only the real server can obtain  $R_A$  from the client's login message  $id || R_A \oplus H(id, pw)$ . After verifying the identity of the client, the server sends a correct response  $R_B || H(K_B, R_A)$  to the client to achieve mutual authentication in Step 2. Due to the discrete logarithm problem and Diffie-Hellman problem, an illegal client cannot compute Diffie-Hellman key  $K_A = K_B = g^{ab} \text{ mod } p$  from  $R_B || H(K_B, R_A)$  and then make a correct response  $id || H(K_A, R_B)$  in Step 3.

**Theorem 5.** *The proposed protected password change protocol can resist a Denial-of-Service attacks.*

*Proof.* In step 1\* of the proposed protected password change protocol, an attacker can replace new password digest  $R_A \oplus H(id, newpw)$  with forged client password digest  $R_A \oplus H(id, newpw) \oplus X$ , in which  $X$  is a random number chosen by the attacker. In Step 2\*, after receiving the replaced messages  $id || R_A \oplus H(id, pw) || R_A \oplus H(id, newpw) \oplus X$ , the server retrieves  $R_A$  from  $R_A \oplus H(id, pw)$  by computing  $R_A \oplus H(id, pw) \oplus (vpw - K) \oplus K$ . Then, the server uses the recovered  $R_A$  to obtain replaced new password verifier  $H(id, newpw) \oplus X$  from  $R_A \oplus H(id, newpw) \oplus X$  by computing  $R_A \oplus H(id, newpw) \oplus X \oplus R_A$ . However, in the proposed protocol, a check item  $H(K_A, R_B, H(id, newpw))$  for new password is added in Step 3\*. The server then updates replaced password verifier  $H(id, newpw) \oplus X$  only if the computed hash value  $H(K_B, R_B, H(id, newpw))$  is equivalent to the received  $H(K_A, R_B, H(id, newpw))$ . But an attacker cannot compute this session key  $K_A = g^{ab} \text{ mod } p$  in hashed value  $H(K_A, R_B, H(id, newpw))$  because the discrete logarithm problem, the Diffie-Hellman problem, and a secure one-way hash function.

**Theorem 6.** *The proposed scheme provides mutual authentication.*

*Proof.* The proposed schemes use the Diffie-Hellman key exchange algorithm [1] to provide mutual authentication; then the key is explicitly authenticated by a mutual confirmation session key,  $K_A = K_B = g^{ab} \text{ mod } p$ .

**Theorem 7.** *The proposed scheme provides the perfect forward secrecy.*

*Proof.* In the proposed schemes, since the Diffie-Hellman key exchange algorithm is used to generate a session key  $K_A = K_B = g^{ab} \text{ mod } p$ , perfect forward secrecy is ensured, as an attacker with a compromised client's password  $pw$  is only able to obtain the  $R_A = g^a \text{ mod } p$  and  $R_B = g^b \text{ mod } p$  from an earlier session. In addition, it is also computationally infeasible to obtain the session key  $g^{ab}$  from  $g^a$  and  $g^b$ , as it is a discrete logarithm problem and Diffie-Hellman problem.

## 7 Conclusions

The current paper demonstrated that Chang et al.'s simple authenticated key agreement and protected password change protocol is vulnerable to stolen-verifier attack and also demonstrated that their protected password change protocol suffers from a Denial-of-Service attacks. We present an improved scheme to isolate such problems. In contrast to Chang et al.'s schemes, the proposed schemes can securely update user passwords without a complicated process, while also providing more security. Therefore, the proposed protocols are more secure than other SAKA-like schemes and protected password change protocols.

## Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments in improving our manuscript. This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment).

## References

1. Diffie, W., Hellman, M.: New Directions in Cryptography. *IEEE Transaction on Information Theory*. Vol. IT-22. No. 6. (1976) 644-654
2. Schneier, B.: *Applied Cryptography-Protocols, Algorithms and Source Code in C*. 2nd edi. John Wiley & Sons Inc. (1995)
3. Seo, D.H., Sweeney, P.: Simple Authenticated Key Agreement Algorithm. *Electronics Letters*. Vol. 35. No. 13. (1999) 1073-1074
4. Bellare, S., Merritt, M.: Encrypted Key Exchange: Password-based Protocols Secure Against Dictionary Attacks. *Proc. of IEEE Conf. on Research in Security and Privacy*. (1992) 72-84
5. Yeh, H.T., Sun, H.M.: Simple Authenticated Key Agreement Protocol resistant to Password Guessing Attacks. *ACM SIGOPS Operation Systems Review*. Vol. 36. No. 4. (2002) 14-22
6. Kobara, K., Imai, H.: Pretty-simple Password-authenticated Key-exchange Protocol Proven to be Secure in the Standard Model. *IEICE Transactions on Fundamentals*. Vol. E85-A. No. 10. (2002) 2229-2237
7. Chang, T.Y., Yang, W.P., Hwang, M.S.: Simple Authenticated Key Agreement and Protected Password Change Protocol. *Computers & Mathematics with Applications*. Vol. 49. No. 5-6. (2005) 703-714
8. Bellare, M., Pointcheval, D., Rogaway, P.: Authenticated Key Exchange Secure Against Dictionary Attack. In *Proc. of EUROCRYPT 2000*. LNCS 1807. (2000) 139-155
9. Lin, C.L., Hwang, T.: A Password Authentication Scheme with Secure Password Updating. *Computers & Security*. Vol. 22. No. 1. (2003) 68-72
10. Menezes, A.J., Oorschot, P.C., Vanstone, S.A.: *Handbook of Applied Cryptography*. CRC Press. New York. (1997)

# A Method for Deciding Quantization Steps in QIM Watermarking Schemes\*

Yunho Lee<sup>1</sup>, Kwangwoo Lee<sup>1</sup>, Seungjoo Kim<sup>1</sup>,  
Dongho Won<sup>1</sup>, and Hyungkyu Yang<sup>2</sup>

<sup>1</sup> Information Security Group, Sungkyunkwan University,  
Suwon-si, Gyeonggi-do, 440-746 Korea  
{younori, kwlee, sjkim, dhwon}@dosan.skku.ac.kr  
<http://www.security.re.kr>

<sup>2</sup> Department of Computer & Media Engineering, Kangnam University,  
Yongin-si, Gyeonggi-do, 449-702 Korea  
[hkyang@kangnam.ac.kr](mailto:hkyang@kangnam.ac.kr)

**Abstract.** In this paper, we propose a method for enlarging quantization steps of a QIM watermarking scheme which determines the perceptual quality and robustness of the watermarked images. In general, increasing the quantization steps leads to good robustness but poor perceptual quality of watermarked images and vice versa. However, if we choose the quantization steps considering the expected quantization results as well as the original images, we can increase both robustness and perceptual quality of the watermarked images.

## 1 Introduction

The advent of the Internet and the wide availability of digital consumer devices such as digital cameras, scanners, and printers make production and distribution of digital contents proliferated. In contrast with analog contents, one can make copy of digital contents without degradation and can tamper without being detected easily. Thus, demanding means for copyright protection is increased rapidly and the digital watermarking is considered as an efficient solution.

Watermarks and watermarking schemes can be divided into various categories in various ways. According to working domain, there are two types of watermarking scheme, spatial domain and frequency domain. The frequency domain schemes are generally considered more robust than the spatial domain schemes and are based on DCT(discrete cosine transform)[3, 4, 13] and DWT(discrete wavelet transform)[12, 14] in general. Various techniques are introduced and applied to watermarking schemes such as spread spectrum[3], SVD(Singular Value Decomposition)[9, 10, 11, 12, 13, 14] and QIM(Quantization Index Modulation)[5, 6, 7, 8, 14].

The QIM watermarking scheme proposed by Chen *et al.* at first, is a blind watermarking model and more robust than spread spectrum or LBM(low-bit

---

\* This work was supported by the University IT Research Center Project funded by the Korea Ministry of Information and Communication.

modulation) one[5]. Though its usefulness and robustness, an attacker can make the watermark undetectable by knowing the quantization steps which are publicly known after proving the existence of watermark.

In this paper, we propose a method for enlarging quantization steps for QIM watermarking scheme which determines the perceptual quality and robustness of the watermarked images. The rest of this paper organized as follows. Section 2 outlines general watermarking model, QIM watermarking scheme, distortion-compensated QIM scheme and Bao's image-adaptive QIM scheme. Section 3 describes the proposed method for deciding quantization steps. Section 4 presents experimental results. This paper is concluded in Section 5.

## 2 Related Works

### 2.1 Watermarking Model

Watermarking is the process that embeds data called a watermark, tag or another image into a multimedia object such that watermark can be detected or extracted later to make an assertion about the object. Let us denote the multimedia object(host signal) by  $x$ , the watermark by  $m$ , the watermarked object by  $y$ , and the extracted(or detected) watermark by  $m'$ , then the general model of watermarking can be depicted as Fig. 1[5].

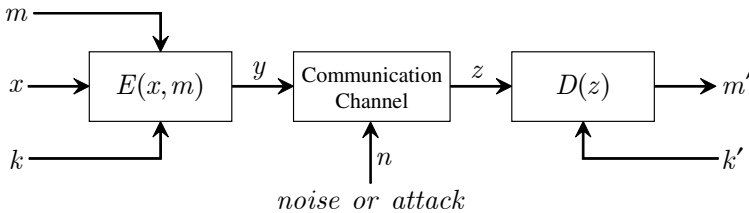


Fig. 1. General Watermarking Model

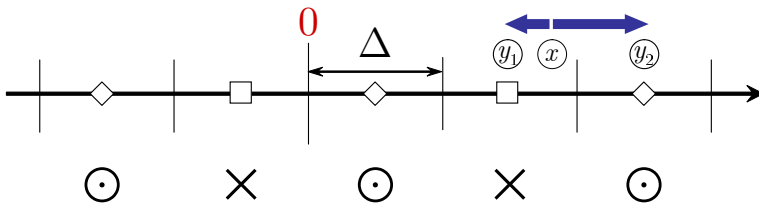
In Fig. 1, the  $n$  can be considered as noise or attack performed by general signal processing such as compression, rotation, resize, crop, and blurring. The  $E(\cdot, \cdot)$  and  $D(\cdot)$  are embedding function and detecting(extracting) function respectively. If the embedding key  $k$  and the detecting key  $k'$  are the same then the scheme is called *symmetric watermarking scheme*. In detecting, if the host signal is required then the scheme is called *non-blind watermarking scheme*.

### 2.2 Quantization Index Modulation

Chen *et al.* developed in 1998 a framework for characterizing the inherent trade-offs between the robustness of the embedding, the distortion to the host signal, and the amount of data embedded and designed a framework of information embedding systems, namely quantization index modulation(QIM), aiming at optimizing the rate-distortion-robustness trade-offs[5]. They developed a method,

the dither modulation, to realize and demonstrate the QIM framework where the embedded information would modulate the dither signal of a dithered quantizer.

QIM embedding methods embed information in the host signal components by quantizing them with a quantizer chosen from an ensemble of quantizers. The watermark  $m$  determines the choice of quantizer. For example if one wishes embed one bit ( $m = 0$  or  $m = 1$ ) in one host signal component of  $x$ , then  $y = q(x, m)$ , where  $q(\cdot, 1)$  and  $q(\cdot, 0)$  are two different quantizers. In Fig. 2  $q(\cdot, 1)$  and  $q(\cdot, 0)$ , depicted as  $\ominus$  and  $\times$  respectively, are uniform, scalar quantizers with step size  $\Delta$ . In this case both the reconstruction points, which are shown as  $\diamond$  and  $\square$  points, and the quantization cells of the two quantizers are shifted versions of each other so the quantizers are dithered quantizers, and this type of QIM is known as dither modulation.



**Fig. 2.** Quantization Index Modulation

If the watermark  $m$  to be embedded is 1 then the  $\ominus$ -quantizer should be used and thus the quantized value(reconstruction point) will be  $y_2$ . In the same way, the quantized value will be  $y_1$  if  $m = 0$ .

Intuitively, properties of the quantizer ensemble can be related directly to the performance parameters of rate, distortion and robustness. For example, the number of quantizers in the ensemble determines the information embedding rate  $r$ . The distance( $\Delta$ ) of reconstruction points determines the embedding induced distortion and robustness.

The general dither modulation scheme is described bellows.

1. Prepare host signal vector  $x = (x_1, \dots, x_n)$ , quantization step vector  $\Delta = (\Delta_1, \dots, \Delta_n)$  and watermark vector  $m = (m_1, \dots, m_n)$ .
2. For  $i = 1, \dots, n$ , repeat the followings.
  - (a) Set  $d = \lfloor \frac{x_i}{\Delta_i} \rfloor$
  - (b) If  $m_i = 1$ , then
    - If  $d \pmod 2 \equiv 0$ , then  $y_i = d\Delta_i + \frac{\Delta_i}{2}$
    - Else if  $(d \leq x_i)$ , then  $y_i = (d + 1)\Delta_i + \frac{\Delta_i}{2}$  else  $y_i = (d - 1)\Delta_i + \frac{\Delta_i}{2}$
  - (c) If  $m_i = 0$ , then
    - If  $d \pmod 2 \equiv 1$ , then  $y_i = d\Delta_i + \frac{\Delta_i}{2}$
    - Else If  $(d \leq x_i)$ , then  $y_i = (d + 1)\Delta_i + \frac{\Delta_i}{2}$  else  $y_i = (d - 1)\Delta_i + \frac{\Delta_i}{2}$ .
3. Publish  $(y_1, \dots, y_n)$  and  $(\Delta_1, \dots, \Delta_n)$ .

### 2.3 Distortion-Compensated Quantization Index Modulation

Although quantization-based methods have been presented since the beginnings of watermarking, it was not until very recently that the idea was revisited from a sound theoretical perspective in the form of a data hiding scheme known as QIM, which hides information by constructing a data-driven set of quantizers. This was later connected to an old paper by Costa to realize that by adding back a fraction of the quantization error, performance could be significantly improved. This scheme was thus termed *distortion compensated QIM*(DC-QIM)[7].

If we denote the information embedding induced distortion(i.e. quantization error) as

$$e = q(x, m) - x, \tag{1}$$

then a fraction of error can be compensated by

$$y = q(x, m) - (1 - \alpha)e, (0 < \alpha \leq 1). \tag{2}$$

Obviously, the DC-QIM can not improve both robustness and perceptual quality, but adjust the robustness-distortion trade-off.

### 2.4 Bao’s Image-Adaptive Watermarking Scheme

Bao *et al.* proposed an image-adaptive watermarking scheme for image authentication by applying a quantization index modulation process on the SVs of the images in wavelet-domain[14]. SVD(Singular Value Decomposition) is a numerical analysis. The most interesting property of SVD for digital watermarking schemes is that the SVs of an image are very stable, that is, when a small perturbation is added to an image, its SVs do not change significantly. For more details on SVD, refer [1, 2]. The scheme is described as follows.

#### Computing Quantization Steps Phase

1. An image  $I = x_0x_1 \cdots x_l$  is transformed into wavelet subbands. In each of the subbands, the coefficients are segmented into blocks  $B_i(i = 1, \dots, n)$ of size  $k \times k$  and SVDs for each of the blocks are computed.
2. Calculate the standard deviation  $\sigma_{B_i}$  and average value  $m_{B_i}$  for DWT coefficients of each block  $B_i$ .
3. Calculate the value  $w_i$  for each block  $B_i$

$$w_i = c_m m_{B_i} + c_\sigma \sigma_{B_i}$$

where  $c_m$  and  $c_\sigma$  are the weight parameters for  $m_{B_i}$  and  $\sigma_{B_i}$ .

4. Calculate  $w_M = \max(w_i)$  and  $w_m = \min(w_i)$  for all  $w_i$ .
5. Compute the quantization step  $\Delta_i$  for block  $B_i$  as

$$\Delta_i = \text{qs}^{\min} + (\text{qs}^{\max} - \text{qs}^{\min}) \frac{w_i - w_m}{w_M - w_m}, \quad i = 1, \dots, n$$

where  $\text{qs}^{\min}$  and  $\text{qs}^{\max}$  are the minimum and maximum quantization step values, respectively, specified by user.

In their experiment, they set  $\text{qs}^{\min} = 9$  and  $\text{qs}^{\max} = 36$  or  $45$ , and let  $c_m = 1.0$ ,  $c_\sigma = 3.0$ .



**Embedding and Extracting Watermark Phase**

1. Compute  $n_v = \|v\| + 1, v = (\lambda_1^i, \dots, \lambda_k^i)$ , where  $v$  is a vector formed by the SVs of each block  $B_i$ .
2. Compute  $S = \lfloor \frac{n_v}{\Delta_i} \rfloor$ , where  $\Delta_i$  is the quantization step for  $n_v$  corresponding to the block  $B_i$  that is computed in the computing quantization steps phase.
3. IF  $(m_i = 1 \wedge S(\bmod 2) \equiv 1)$ , then  $S = S + 1$   
 IF  $(m_i = 0 \wedge S(\bmod 2) \equiv 0)$ , then  $S = S + 1$
4. Compute the value  $n'_v = \Delta_i S + \frac{\Delta_i}{2}$  and the modified SV

$$(\gamma_1^i, \dots, \gamma_k^i) = (\lambda_1^i, \dots, \lambda_k^i) \times \frac{n'_v}{n_v}.$$

5. Reconstruct the blocks and watermarked image using the modified SVs.

Let  $\tilde{B}_i$  be a block with an embedded watermark bit, the extraction of the watermark can be described as follows.

1. Segment the watermarked image into blocks  $\tilde{B}_i (i = 1, \dots, n)$  of size  $k \times k$  after wavelet transform.
2. Compute the value  $\tilde{n}_v = \|u\| + 1, u = (\gamma_1^i, \dots, \gamma_k^i)$ , where  $u$  is a vector formed by the SVs of each block  $\tilde{B}_i$ .
3. Compute  $S = \lfloor \frac{\tilde{n}_v}{\Delta_i} \rfloor$ .
4. IF  $S(\bmod 2) \equiv 0$ , then the embedded bit is 1. Otherwise, it is 0.

**3 Proposed Method**

**3.1 Basic Ideas**

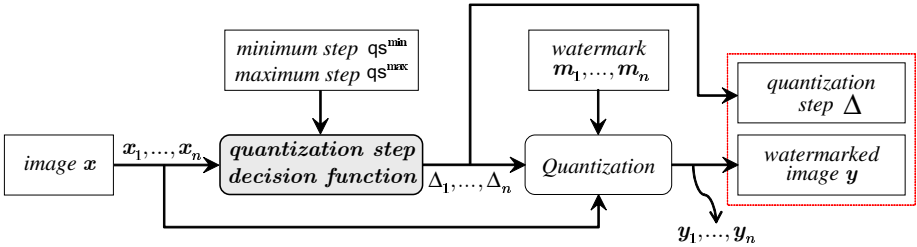
One desires a watermarking scheme to have high rate, low distortion, and high robustness, but in general these three goals tend to conflict. Thus, the performance of an information embedding system is characterized in terms of its achievable rate-distortion-robustness trade-offs.

Let's consider distortion and robustness among those three aspects. Higher robustness means larger quantization steps used and induces higher distortion, i.e. quantization error. To tackle this robustness-distortion tradeoff, Bao *et al.* proposed image-adaptive decision method for quantization steps which is better than using constant steps. We propose a different decision method for quantization steps considering the expected quantization results as well as the host signal in order to increase robustness and decrease distortion. Fig. 3 shows (a) the Bao's method and (b) the proposed method respectively. Moreover, Bao's quantization method is not efficient because the  $x_i$ 's distortion range is  $0 - \frac{3\Delta_i}{2}$  while in case of general dither modulation it ranges from  $0 - \Delta_i$  [5, 6, 7]. The comparison of Bao's quantization method and the binary dither modulation is depicted in Fig. 4.

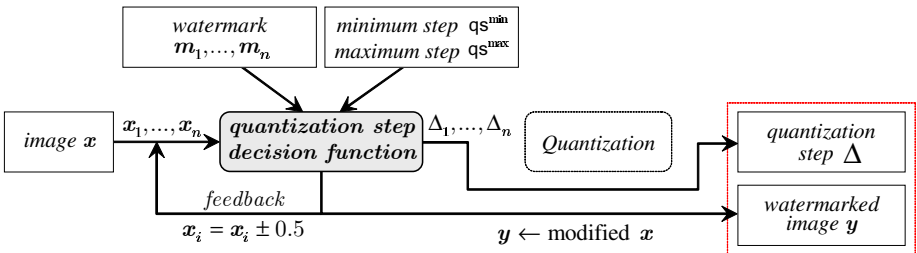
Let  $x_i$  be the  $i$ -th positive integer source signal,  $q_i$  be the  $i^{th}$  quantizer with quantization step  $\Delta_i$ , and  $y_i$  be the watermarked signal. If we use binary dithered modulation, the  $\Delta_i$  can be chosen from one of the two sets  $\Delta_{m=1}$  and  $\Delta_{m=0}$ .

$$\Delta_{m=0} = \left\{ \frac{2y_i}{2c+1} \right\}, \text{ where } c = 1, 3, \dots \tag{3}$$

$$\Delta_{m=1} = \left\{ \frac{2y_i}{2c+1} \right\}, \text{ where } c = 0, 2, \dots \tag{4}$$

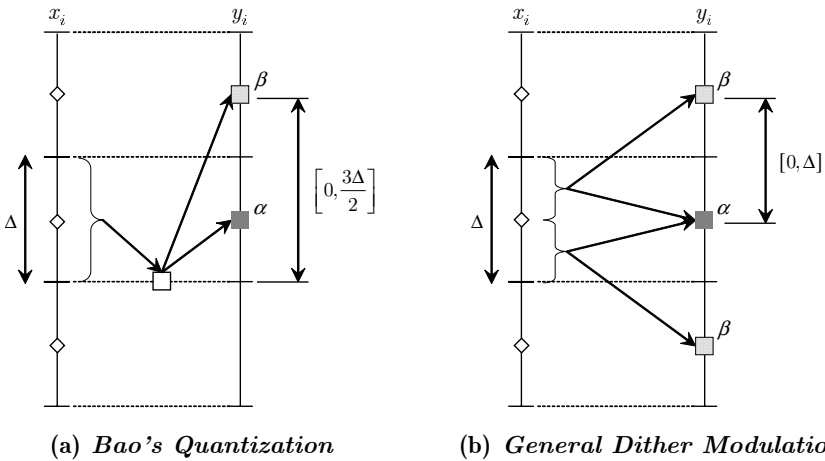


(a) Bao's quantization steps decision and watermarking method



(b) Proposed quantization steps decision and watermarking method

Fig. 3. The two methods for deciding quantization steps. (a) Bao's method. (b) Proposed method.



(a) Bao's Quantization

(b) General Dither Modulation

Fig. 4. The comparison of Bao's quantization method and the general dither modulation

**Table 1.** The possible quantization steps for  $x_i = 351$

$m = 1$						$m = 0$							
$c = 0$	$c = 2$	$c = 4$	$c = 6$	$c = 8$	$c = 10$	...	$c = 1$	$c = 3$	$c = 5$	$c = 7$	$c = 9$	$c = 11$	...
702	140.4	78	54	41.29	33.43	...	234	100.29	63.82	46.8	36.95	30.52	...

Thus, if we set  $y_i = x_i$  and can find appropriate integer  $\Delta_i$ , the embedding induced distortion will be zero. For example, given  $x_i = 351$  the possible quantization steps are shown in the following Table 1.

Intuitively,  $\max_{c=0,2,\dots} \left(\frac{2y_i}{2c+1}\right)$  is  $2y_i$  and  $\max_{c=1,3,\dots} \left(\frac{2y_i}{2c+1}\right)$  is  $\frac{2}{3}y_i$ . If we assume that only integer values can be quantization steps,  $\Delta_{m=1}$  has at least one integer value whereas  $\Delta_{m=0}$  does not. The algorithm for finding integer quantization steps is described as follows.

1. select maximum and minimum quantization step  $qs^{\max}$  and  $qs^{\min}$  respectively and distortion threshold  $t$ .
2. set  $x' \leftarrow x$ .
3. if  $m = 0$ , for  $c = 0, 2, 4, \dots$ , compute integer quantization step  $qs$  as

$$qs^{\min} \leq qs_i = \max_{\forall c} \left(\frac{2x'_i}{2c+1}\right) \leq qs^{\max}. \tag{5}$$

4. if  $m_i = 1$ , for  $c = 1, 3, 5, \dots$ , compute integer quantization step  $qs_i$  as (5).
5. if no quantization step is found, increase or decrease  $x'_i$  by 0.5.
6. go to step 3, until  $|x'_i| > |x_i| + t$ .
7. if  $|x'_i| > |x_i| + t$  then set  $x' \leftarrow x$ .
8. set  $x \leftarrow x'$ .

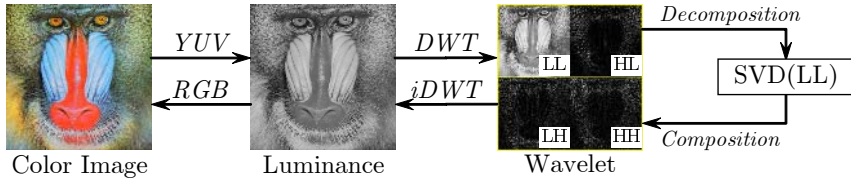
### 3.2 Deciding the Quantization Step Range

Intuitively, proper selection of the range of the quantization step values [ $qs^{\min}$ ,  $qs^{\max}$ ] is important to achieve high robustness and low distortion. Besides robustness and distortion, transmission overhead should be considered for selecting the range of quantization step values.

If we set the range as broad as possible, i.e.  $[1, 2 \max(x_i)]$ , then we can achieve maximum robustness and minimum distortion. However, broader range requires more data size of quantization parameters. Thus, it is necessary to choose a proper range of quantization step values considering the data size of quantization parameters. In this paper, we will use  $[1, 255]$  as the practical range of quantization step values.

## 4 Experimental Results and Performance Comparison

For our experiments, we use a general watermarking scheme based on DWT (discrete wavelet transform) and SVD(singular value decomposition). The scheme is described briefly as follows(see Fig. 5).



**Fig. 5.** Watermarking Procedure for Experiments

1. Perform DWT transform on grayscale image. If a color image is presented in RGB then it can be converted to the corresponding luminance matrix as

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} +0.299 & +0.587 & +0.144 \\ -0.148 & -0.289 & +0.437 \\ +0.615 & -0.515 & -0.1 \end{pmatrix} \times \begin{pmatrix} R \\ G \\ B \end{pmatrix}. \tag{6}$$

2. Segment the image of LL band into blocks  $B_i$  of size  $4 \times 4$ ,  $i = 1, \dots, (\frac{N}{8})^2$ , where  $N$  is the width and height of the source image.
3. Compute host signal  $x_i = |v_i|$ , where  $v_i = (\lambda_1^i, \lambda_2^i, \lambda_3^i, \lambda_4^i)$ ,  $v_i$  is a vector formed by SVs  $\Lambda$  of each block  $I_i$ .

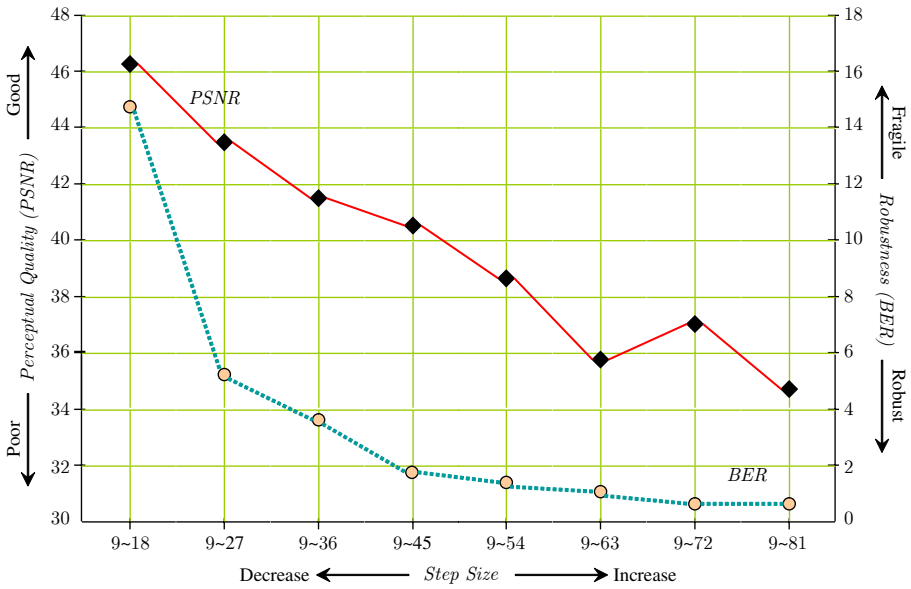
$$B_i = U\Lambda V^T.$$

4. Compute watermarked signal  $y_i = E(x_i, \cdot)$  and modify  $v_i$  as  $v'_i = v_i \times (y_i/x_i)$ .
5. Perform iDWT transform on the watermarked signal and convert it to RGB color space if necessary.

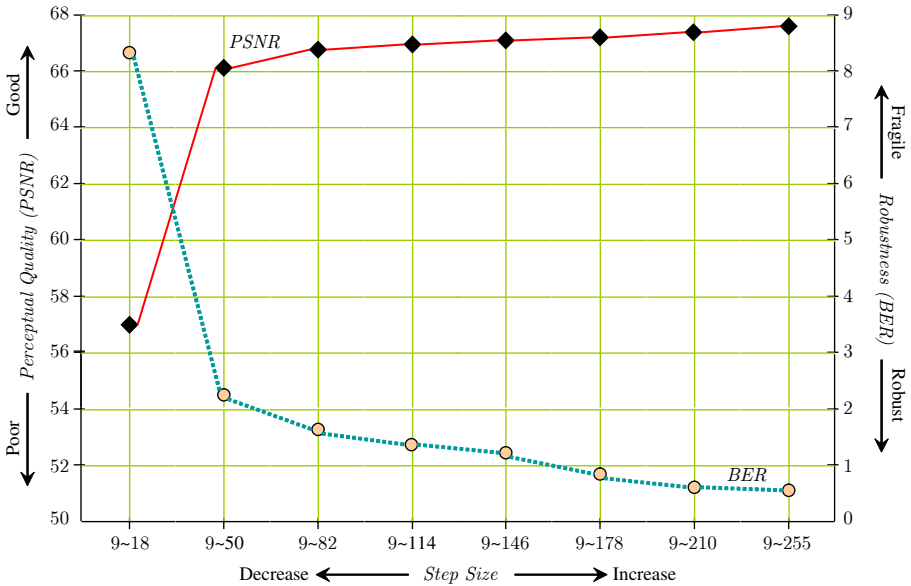
The PSNR of the watermarked image is about 67~68 which is extremely high in compared with the Bao’s QIM watermarking scheme(see Table 2). Moreover, the robustness of the proposed scheme is better than the Bao’s scheme which can be measured by BER(Bit-Error Ratio). It is generally accepted that higher perceptual quality means lower robustness and vice versa. However, using our scheme both good perceptual quality and robustness can be achieved at the same time.

**Table 2.** Performance comparison of Bao’s and the proposed scheme in perceptual quality and robustness

Image	Scheme	Perceptual Quality (PSNR)	Robustness(BER)			Step Size
			JPEG(50%)	JPEG(10%)	Median(4x4)	
Lena	Bao	41.63	3.857	24.457	31.152	9~36
	<b>Proposed</b>	<b>67.95</b>	<b>1.758</b>	<b>7.373</b>	<b>13.379</b>	<b>1~255</b>
Baboon	Bao	40.73	0.415	16.016	42.944	9~36
	<b>Proposed</b>	<b>68.08</b>	<b>0.903</b>	<b>7.251</b>	<b>19.873</b>	<b>1~255</b>
Peppers	Bao	41.28	3.223	20.288	28.633	9~36
	<b>Proposed</b>	<b>68.33</b>	<b>2.222</b>	<b>7.813</b>	<b>13.623</b>	<b>1~255</b>



(a) Bao's Watermarking Scheme



(b) Proposed Watermarking Scheme

Fig. 6. The relations between the step sizes and the PSNR/BER. The dotted line indicates the BER and the solid line indicates the PSNR.

Intuitively the robustness of a QIM watermarking scheme is determined by the quantization steps  $\Delta_i$  as shown in Fig. 6. For Bao's scheme, increasing the step size causes the watermarked images to have poor perceptual quality while good robustness. However, for our scheme, increasing the step size causes the watermarked images to have both good perceptual quality and robustness.

## 5 Conclusion

In this paper, a method for deciding quantization steps for QIM watermarking schemes is presented. It is shown that if we choose quantization steps considering the expected quantization results as well as host signal, we can increase the quantization steps and achieve both good robustness and perceptual quality. Also we presented experimental results of robustness(BER) and perceptual quality(PSNR) in case of two different ranges  $[1, 255]$  and  $[1, 2 \max(x_i)]$ . While choosing the range is depends on the various applications and requirements such as minimum robustness, maximum distortion, and data size of quantization parameters, the range  $[1, 255]$  is a good choice because we can achieve good PSNR, BER, and data size in compared with the range  $[1, 2 \max(x_i)]$ .

## References

1. H.C.Andrews and C.L.Patterson, "Singular Value Decomposition(SVD) Image Coding," *IEEE Trans. on Communications*, vol.COM-24, pages 425-432, 1976
2. G.H.Golub and C.Reinsch, "Singular Value Decomposition and Least Squares Solutions," *Nun. Math.*, vol.14, pages 403-420, 1970
3. I.J.Cox, J.Killian, T.Leighton, and T.Shamoon, "Secure Spread Spectrum Watermarking for Multimedia," *IEEE Trans. on Image Processing*, vol.6, no.12, pages 1673-1687, 1997
4. M.D.Swanson, B.Zhu, and A.H.Tewfek, "Transparent Robust Image Watermarking," *Proc. of IEEE International Conference on Image Processing*, 1996
5. B.Chen and G.W.Wornell, "Digital Watermarking and Information Embedding Using Dither Modulation," *Proc. of IEEE MMSP-98*, pages 273-278, 1998
6. B.Chen and G.W.Wornell, "Dither Modulation: A New Approach to Digital Watermarking and Information Embedding," *Proc. of SPIE: Security and Watermarking of Multimedia Contents*, pages 342-353, 1999
7. B.Chen and G.W.Wornell, "Quantization Index Modulation: A Class of Provably Good Methods for Digital Watermarking and Information Embedding," *IEEE Trans. on Information Theory*, vol.47, no.4, pages 1423-1443, 2001
8. B.Chen and G.W.Wornell, "Quantization Index Modulation Methods for Digital Watermarking and Information Embedding of Multimedia," *Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology, Special Issue on Multimedia Signal Processing*, vol.27, no.1-4, pages 7-33, 2001
9. V.I.Gorodetski, L.J.Popyack, and V.Samoilov, "SVD-based Approach to Transparent Embedding Data into Digital Images," *Proc. Int. Workshop MMM-ACNS*, pages 263-274, 2001
10. R.Liu and T.Tan, "An SVD-based Watermarking Scheme for Protecting Rightful Ownership," *IEEE Trans. on Multimedia*, vol.4, no.1, pages 121-128, 2001

11. S.Byun, S.Lee, A.H.Tewfik and B.Ahn, "A SVD-Based Fragile Watermarking Scheme for Image Authentication," *Digital Watermarking, First International Workshop, IWDW 2002*, LNCS 2613, pages 170–178, 2002.
12. E.Gamic and A.M.Eskicioglu, "Robust DWT-SVD Domain Image Watermarking: Embedding Data in All Frequencies," *Proc. of MM&Sec04*, 2004
13. F.Huang and Z.H.Guan, "A Hybrid SVD-DCT Watermarking Method Based on LPSNR," *Pattern Recognition Letters*, vol.25, no.15, pages 1769–1775, 2004
14. P.Bao and X.Ma, "Image Adaptive Watermarking Using Wavelet Domain Singular Value Decomposition," *IEEE Trans. on Circuits and Systems for Video Technology*, vol.15, no.1, pages 96–102, 2005

# A New DDoS Detection Model Using Multiple SVMs and TRA\*

Jungtaek Seo<sup>1</sup>, Cheolho Lee<sup>1</sup>, Taeshik Shon<sup>2</sup>, Kyu-Hyung Cho<sup>2</sup>,  
and Jongsub Moon<sup>2</sup>

<sup>1</sup> National Security Research Institute,  
62-1 Hwaam-dong, Yuseong-gu, Daejeon 305-348, Republic of Korea  
{seojt, chlee}@etri.re.kr

<sup>2</sup> CIST, Korea University,  
1-Ga, Anam-dong, Sungbuk-Gu, Seoul, Republic of Korea  
{743zh2k, mathbank, jsmoon}@korea.ac.kr

**Abstract.** Recently, many attack detection methods adopts machine learning algorithm to improve attack detection accuracy and automatically react to the attacks. However, the previous mechanisms based on machine learning have some disadvantages such as high false positive rate and computing overhead. In this paper, we propose a new DDoS detection model based on multiple SVMs (Support Vector Machine) in order to reduce the false positive rate. We employ TRA (Traffic Rate Analysis) to analyze the characteristics of network traffic for DDoS attacks. Experimental results show that the proposed model is a highly useful classifier for detecting DDoS attacks.

## 1 Introduction

As we can see in the incidents of Distributed Denial of Service (DDoS) attacks against commercial web sites such as *Yahoo*, *e-Bay*, and *E\*Trade*, computing resources connected to the Internet are vulnerable to DDoS attacks [1], [2], [3]. DDoS attacks can temporarily disable the network services or damage systems by flooding a huge number of network packets for several minutes or longer.

Since these DDoS attacks are harmful to almost all networked systems which have limited computing resources (e.g. network bandwidth, memory, CPU, etc), these attacks are regarded as a serious problem, and thus much research is in progress to detect and prevent them [4], [5], [6].

In our earlier research, we presented Traffic Rate Analysis (TRA) to analyze the characteristics of network traffic for the DDoS attacks [7], [8], [9]. TRA is a network traffic analyzing method which examines the occurrence rate of a specific type of packet within the stream of monitored network traffic and is composed of a TCP flag rate and a Protocol rate. The result of analyzing network traffic using TRA showed us that there are distinct and predictable differences between normal traffic and DDoS

---

\* This work was supported by the Ministry of Information Communication, Korea, under the Information Technology Research Center Support Program supervised by the IITA.



attack traffic. We were able to generate DDoS detection rules by compiling the experimental results with a SVM [10]. However, the false positive rate of the model using single SVM is too high. In order to reduce the false positive rate and to increase the detection rate, we propose the model based on multiple SVMs instead of single one. The experimental results show the proposed detection method has high degree of performance, and detects various DDoS attacks successfully with low false positive rate.

We introduce related research in section 2, and explain TRA in section 3. The background knowledge of SVM is discussed in section 4. In section 5, the experimental environment is introduced and the detection performance of SVM and other machine learning algorithms are tested and compared. Lastly, we mention the conclusion of this research and the direction of future work in section 6.

## 2 Related Work

Detecting the DDoS attacks is an essential step to defend DDoS attacks. Thus, there have been many researches to detect the DDoS attacks [4], [5], [6]. When DDoS attacks occur, there is a big mismatch between the packet flows “to-rate” toward the victim and “from-rate” from the victim. Gil and Poletto propose the method that examines the disproportion between “to-rate” and “from-rate” in order to detect DDoS attacks [4]. Kulkarni et al [5] presents DDoS detection methods based on randomness of IP spoofing. Almost DDoS attackers use IP spoofing to hide their real IP addresses and locations. Since spoofed IP addresses are generated randomly, this characteristic of randomness may be used to reveal the occurrence of DDoS attacks. Kulkarni’s method uses *Komogorov complexity metrics* to measure the randomness of source IP addresses in network packet headers [11]. Wang et al. proposed the method that detects DDoS attack based on the protocol behavior of *SYN-FIN(RST)* pairs [6]. In the normal situation, the ratio of *SYN* and *FIN* is balanced because of the characteristic of the TCP 3-Way handshake. However, the ratio of *SYN* packet increases drastically during the SYN flooding attack. By monitoring sudden change of the ratio of *SYN* and *FIN*, the method detects SYN flooding attacks.

However, these approaches are based on the specific characteristics of the attacks such as mismatch of “to-rate” and “from-rate”, effect of IP spoofing, and unbalance of the ratio of *SYN* and *FIN* packet. Thus, these may not properly detect the attack that use undefined characteristic. For example, Gil’s method is not applicable to detect attacks using IP spoofing since the method cannot discriminate legitimated packet and spoofed packet, and Wang’s method is only applicable to SYN flooding attacks. On the other hand, the proposed detection model automatically generates detection rules using TRA and multiple SVM.

## 3 Traffic Rate Analysis

### 3.1 Definition of Traffic Rate Analysis

Traffic rate analysis was defined as measuring packet traffic in a network [7]. It examines the occurrence rate of a specific type of packets within the stream of moni-

tored network traffic, and is composed of TCP flag rate and Protocol rate. TCP flag rate is defined in the following equation.

$$R_{td}[F\ i\ |o] = \frac{\sum \text{flag}(F) \text{ in a TCP header}}{\sum \text{TCP packets}} \tag{1}$$

TCP flag rate means the ratio of the number of a specific TCP flag to the total number of TCP packets. In the equation (1), a TCP flag 'F' can be one of *SYN*, *FIN*, *RST*, *ACK*, *PSH*, *URG*, and *NULL*, and 'td' is the time interval used to calculate the value. The direction of network traffic is expressed as 'i' (inbound) and 'o' (outbound). For example,  $R_i[Si]$  means the occurrence rate of *SYN* flags within TCP packets when measuring inbound network traffic (toward the monitored network) during interval 1.

$$R_{td}[[TCP|UDP|ICMP]i\ |o] = \frac{\sum [TCP|UDP|ICMP] \text{ packets}}{\sum IP \text{ packets}} \tag{2}$$

Protocol rate is defined in equation (2). It means the ratio of specific Transport-Layer protocol (e.g. TCP, UDP, and ICMP) packets to total Network-Layer (IP) protocol packets. For instance,  $R_i[TCPi]$  means the occurrence rate of TCP packets within IP packets when measuring outbound network traffic (from the monitored network) during interval 1.

### 3.2 Network Traffic Changes Under DDoS Attacks

To analyze the change of network traffic from normal web traffic to DDoS attack traffic or vice versa, it is necessary to make a network environment truly identical with the real Internet environment.

Our experimental target is web traffic, and web traffic is composed of HTTP requests and replies based on TCP sessions. For example, when a user clicks a certain web site address on his or her web browser, the web browser establishes TCP connections to the relevant web server. After that, the web browser sends HTTP requests to the web server, and the web server sends HTTP replies to the web browser.

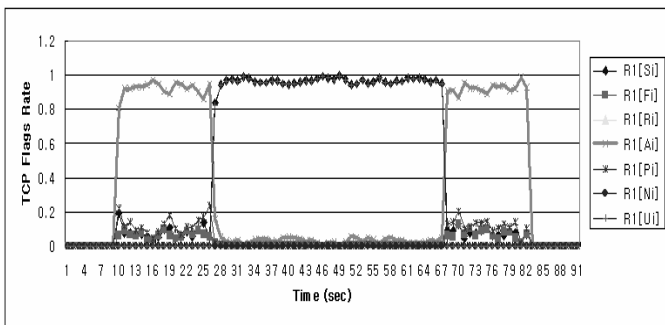


Fig. 1. Network traffic under SYN flooding attack

Since web service is based on TCP connection, the number of HTTP requests in a TCP session (*R/C*: Requests per connection) and the number of TCP sessions simultaneously established (*SC*: Simultaneous Connection) are the key features of web traffic in terms of network traffic analysis. In other words, we can simulate various web traffic environments by adjusting these two features (*R/C* and *SC*).

*R/C* values include 1, 2, 5, and 10, and *SC* can take on values of 5, 10, 50, 100, 150, and 200. Thus we have twenty-four different network environments. With these various web traffic settings, we compared normal web traffic with DDoS attack traffic.

Fig. 1 shows us that  $R1[Si]$  and  $R1[Ui]$  drastically change (go up to almost 1.0) and the other flags decrease (almost 0.0) relatively under SYN flooding attack. When web traffic flows from the 9th second to the 83rd second, a SYN flooding attack occurs between the 26th and 67th second. This phenomenon is caused by the burst of SYN and URG packets, which are generated by SYN flooding attack.

Furthermore, we can also see big changes of network traffic during other types of DDoS attacks such as ICMP flooding attacks or UDP flooding attacks [7], [8], [9].

## 4 Support Vector Machine

### 4.1 Background

Support Vector Machine (SVM) is a learning machine that plots the training vectors in high-dimensional feature space, and labels each vector by its class. SVM views the classification problem as a quadratic optimization problem. It combines generalization control with a technique to avoid the “curse of dimensionality” by placing an upper bound on a margin between the different classes, making it a practical tool for large and dynamic data sets. SVM classifies data by determining a set of support vectors, which are members of the set of training inputs that outline a hyper plane in feature space. The SVM is based on the idea of structural risk minimization, which minimizes the generalization error, i.e. true error on unseen examples. The number of free parameters used in the SVM depends on the margin that separates the data points to classes but not on the number of input features. Thus SVM does not require a reduction in the number of features in order to avoid over fitting. SVM provides a generic mechanism to fit the data within a surface of a hyper-plane of a class through the use of a kernel function. The user may provide a kernel function, such as a linear, polynomial, or sigmoid curve, to the SVM during the training process, which selects support vectors along the surface of the function. This capability allows classifying a broader range of problems [12], [13].

### 4.2 SVM for Categorization

In this section we review some basic ideas of SVM. Given the training data set  $\{(x_i, d_i)\}_{i=1}^N$  with input data  $x_i \in R^N$  and corresponding binary class labels  $d_i \in \{-1, 1\}$ , the SVM classifier formulation starts from the following assumption. The classes represented by the subset  $d_i = 1$  and  $d_i = -1$  are linearly separable, where  $\exists w \in R^N$ ,  $b \in R$  such that

$$\exists w, b \quad s.t \quad \begin{cases} w^T x_i + b > 0 & \text{for } d_i = +1 \\ w^T x_i + b < 0 & \text{for } d_i = -1 \end{cases} \tag{3}$$

The goal of SVM is to find an optimal hyper plane for which the margin of separation,  $\rho$ , is maximized.  $\rho$  is defined by the separation between the separating hyperplane and the closest data point. If the optimal hyperplane is defined by  $(w_0^T \cdot x) + b_0 = 0$ , then the function  $g(x) = w_0^T \cdot x + b_0$  gives a measure of the distance from  $x$  to the optimal hyperplane.

Support Vectors are defined by data points  $x^{(s)}$  that lie the closest to the decision surface. For a support vector  $x^{(s)}$  and the canonical optimal hyperplane  $g$ , we have

$$r = \frac{g(x^s)}{\|w_0\|} = \begin{cases} +1/\|w_0\| & \text{for } d^{(s)} = +1 \\ -1/\|w_0\| & \text{for } d^{(s)} = -1 \end{cases} \tag{4}$$

Since, the margin of separation is  $\rho \propto \frac{1}{\|w_0\|}$ .  $\|w_0\|$  should be minimal to achieve the maximal separation margin. Mathematical formulation for finding the canonical optimal separation hyperplane, given the training data set  $\{(x_i, d_i)\}_{i=1}^N$ , solves the following quadratic problem

$$\left\{ \begin{array}{l} \min \tau(\omega, \xi) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l \xi_i \\ s.t \quad d_i(w^T x_i + b) \geq 1 - \xi_i \quad \text{for } \xi_i \geq 0, \quad i = 1, \dots, l \end{array} \right\} \tag{5}$$

Note that the global minimum of above problem must exist, because  $\Phi(w) = \frac{1}{2} \|w_0\|^2$  is convex in  $w$  and the constrains are linear in  $w$  and  $b$ . This constrained optimization problem is dealt with by introducing Lagrange multipliers  $a_i \geq 0$  and a Lagrangian function given by

$$L(w, b, \zeta, a, v) = \tau(w, \zeta) - \sum_{i=1}^l a_i [d_i(w_i^T x_i + b) - 1 + \zeta_k] - \sum_{i=1}^l v_i \zeta_i \tag{6}$$

which leads to

$$\frac{\partial L}{\partial w} = 0 \Leftrightarrow w - \sum_{i=1}^l a_i d_i x_i = 0 \quad (\because w = \sum_{i=1}^l a_i d_i x_i) \tag{7}$$

$$\frac{\partial L}{\partial b} = 0 \Leftrightarrow \sum_{i=1}^l a_i d_i = 0 \tag{8}$$

The solution vector thus has an expansion in terms of a subset of the training patterns, namely those patterns whose  $a_i$  is non-zero, called Support Vectors. By the Karush-Kuhn-Tucker complementarity conditions, we have,

$$a_i [d_i (w^T x_i + b) - 1] = 0 \quad \text{for } i = 1, \dots, N \tag{9}$$

by substituting (7), (8) and (9) into equation (6), find multipliers  $a_i$  for which

$$\max \Theta(a) = \sum_{i=1}^l a_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l a_i a_j d_i \langle x_i \cdot x_j \rangle \tag{10}$$

$$s.t. \quad 0 \leq a_i \leq c, \quad i = 1, \dots, l \quad \text{and} \quad \sum_{i=1}^l a_i y_i = 0 \tag{11}$$

The hyperplane decision function can thus be written as

$$f(x) = \text{sgn} \left( \sum y_i a_i \cdot (x \cdot x_i) + b \right) \tag{12}$$

where  $b$  is computed using (9).

To construct the SVM, the optimal hyperplane algorithm has to be augmented by a method for computing dot products in feature spaces nonlinearly related to input space. The basic idea is to map the data into some other dot product space (called the feature space)  $F$  via a nonlinear map  $\Phi$ , and to perform the above linear algorithm in  $F$ , i.e. nonseparable data  $\{(x_i, d_i)\}_{i=1}^N$ , where  $x_i \in R_N$ ,  $d_i \in \{+1, -1\}$ , preprocess the data with,

$$\Phi: R^N \rightarrow \Theta(x) \quad \text{where} \quad N \ll \dim(F) \tag{13}$$

Here  $w$  and  $x_i$  are not calculated. According to Mercer's theorem,

$$(\Phi(x_i) \cdot \Phi(x_j)) = K(x_i, x_j) \tag{14}$$

and  $K(x, y)$  can be computed easily on the input space. Finally the nonlinear SVM classifier becomes

$$f(x) = \text{sgn} \left( \sum_{i=1}^l a_i d_i K(x_i \cdot x) + b \right) \tag{15}$$

## 5 Experiment

### 5.1 DDoS Detection Process

Fig. 2 shows the overall composition of the DDoS detection process.

It is composed of two steps. One is the preprocessing step, and the other is the training and testing step. In the preprocessing step, it captures raw network traffic from both DDoS and legitimate network traffics, and extracts features from the captured raw network traffic using TRA method for each training and test set. For both training and testing, we used 10 features;  $R_1[S_i]$ ,  $R_1[F_i]$ ,  $R_1[R_i]$ ,  $R_1[A_i]$ ,  $R_1[P_i]$ ,  $R_1[U_i]$ ,  $R_1[N_i]$ ,  $R_1[TCP_i]$ ,  $R_1[UDP_i]$ , and  $R_1[ICMP_i]$ .

In the training and testing step, they are trained by each machine using the training set. To train the machine, we classify input packets of the training set as *attack* (-1) and *normal* (+1). Normal web traffic was categorized as *normal*, and the various DDoS attack traffic was categorized as *attack*. The trained machines evaluate test sets, and discriminate legitimate traffic and DDoS traffic. In the experiments, we used

two different machine learning models; single SVM and multiple SVMs model. Multiple SVMs model consists of several SVMs that are learned by different training data, and each SVM is specialized to specific attacks (e.g., Smurf attack, Tfn2k attack, SYN flooding). In the experiment, we categorized the attacks into three types; DoS attack, DDoS attack, and DrDoS attack.

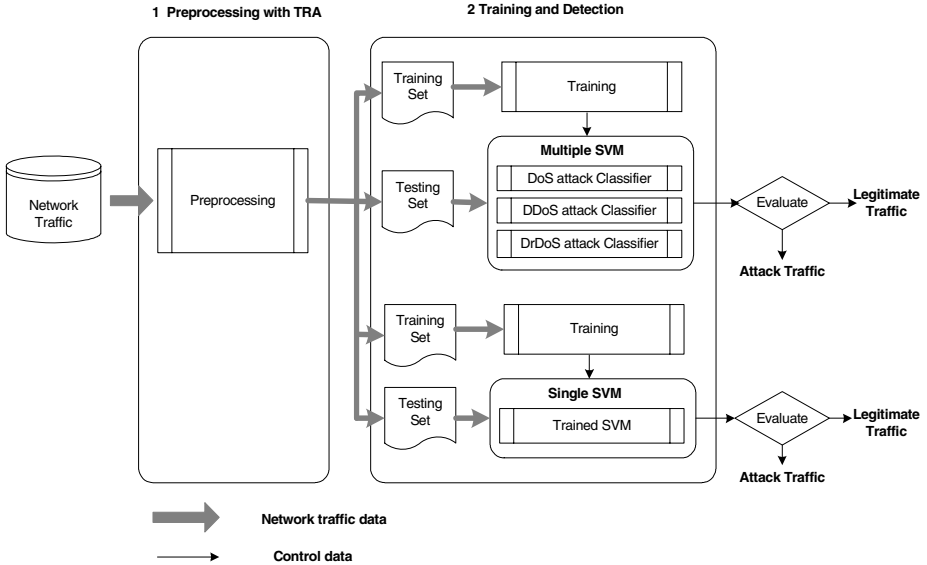


Fig. 2. Overall composition of DDoS detection process

### 5.2 Experimental Environment

Our traffic monitor was developed with the network packet capturing library *libpcap*. It is divided into two modules. One is the module for capturing network traffic and calculating TCP flag rate and protocol rate from the monitored network, and the other is the DDoS attack detection module tuned by the support vector machine. The TRA analyzer is located on the adjacent site of a target Web server and captures both inbound and outbound network traffic packets through an Ethernet hub, and then calculates TCP flag rate and protocol rate in every second.

Web clients are composed of four hosts using SPECweb99 to generate normal web traffic toward an Apache web server. To generate DDoS attack traffic toward the web server we used several attack tools as shown in the Table 1. We collected network traffic for 100 seconds during training and testing time. Web clients continually generated normal web traffic toward the web server, and various attacks occurred between the 25th and 75th seconds.

In the multiple SVMs model, each attack classifier is trained by own training sets (e.g., DoS training set, DDoS training set, and DrDoS training set), and each training set consists of 500 normal data and 500 attack data. On the other hand, the training data set of single SVM model consists of 1500 normal data and 1500 attack data, and the attack data is not classified. Table 2 shows the composition of training data sets.

**Table 1.** Classification of Attack Tools

Attack tool	Attacks
Targa3 (DoS)	bonk, jolt, land, nestea, newtear, syndrop, teardrop, winnuke, sai hyousen, oshare, etc.
TFN2K (DDoS)	ICMP flooding, UDP flooding, etc.
pHorgam (DrDoS)	DrDoS

**Table 2.** Composition of training data sets

Model	Classifier	Data Type	Number of Data
Multiple SVM	DoS attack classifier	DoS attack	500
		Normal	500
	DDoS attack classifier	DDoS attack	500
		Normal	500
	DrDoS attack classifier	DrDoS attack	500
		Normal	500
Single SVM		Attack	1500
		Normal	1500

### 5.3 Detection Performance Analysis

For each training set, we used *dot* and *polynomial* kernel with epsilon 0.01. We used 1000 as capacity parameter and +0.01 as epsilon parameter. The detection performance using multiple SVM and single is shown in Table 3.

**Table 3.** Detection performance of multiple SVM and single SVM

Kernel	Model	False Positive (%)	False Negative (%)		
			DoS	DDoS	DrDoS
Poly	Multiple SVM	26.82	4.33	0.80	0.19
	SVM	40.15	5.32	1.60	1.38
Linear	Multiple SVM	9.84	0.19	1.40	1.36
	SVM	20.47	2.95	2.60	8.87

As we can see in Table 3, multiple SVMs show slightly higher detection performance than single SVM with decrease of false positive rate. Since SVM is a binary classifier, the normal region decreases according to increasing of attack region. Decreasing of normal region means that the increasing of probability of false positive. Fig. 3 shows the relation between normal region and attack region. Each attack type (e.g., DoS, DrDoS, and DDoS) has own attack region. In the experiment, single SVM model merged these regions into a single huge attack region, while multiple SVMs model does not merge these attack regions. The reason is that multiple SVMs trains each attack classifier independently using different training data. Thus, the false positive rate of multiple SVM model is lower than single the rate of single SVM model.

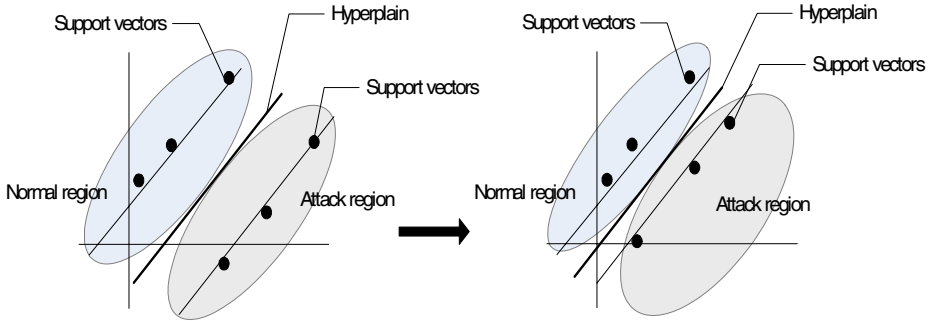


Fig. 3. Relation between normal region and attack region in SVM

## 6 Conclusions

In this paper, we utilized a TRA (Traffic Rate Analysis) method proposed in earlier research [7], [8]. In addition, we adopted multiple SVMs (Support Vector Machine) model instead of single SVM model in order to compile detection rules. As a result, multiple SVMs model shows slightly higher detection accuracy and lower false positive rate. We expect that our approach will be useful in providing early detection of DDoS attacks against the Internet infrastructure. However, our machine learning scheme does not have unsupervised feature but supervised feature. It may mean sometimes if our scheme meet a kind of unexpected situation, it is difficult it can work well or not. Thus, we need additional work using unsupervised learning method without pre-existing knowledge. In our future work, we will use other kernel function methods of SVM and various machine learning methods. Moreover, we are going to focus on detecting and defending against other types of attacks like worms.

## References

1. Garber, L.: Denial-of-Service Attacks Rip the Internet, *IEEE Computer*, vol. 33(4), (2000) 12-17
2. Houle, J.K., and Weaver, M.G.: Trends in Denial of Service Attack Technology, CERT Coordination Center, (2001)
3. Moore, D., Voelker, G.M., and Savage, S.: Inferring Internet Denial-of-Service Activity. In *Proceedings of the 10th USENIX Symposium*, (2001) 9-22
4. Gil, T.M, and Poletto, M.: MULTOPS: a data-structure for bandwidth attack detection, In *Proceedings of the 10th USENIX Security Symposium*, (2001) 23-38
5. Kulkarni, A.B., Bush, S.F., and Evans, S.C.: Detecting Distributed Denial-of-Service Attacks Using Kolmogorov Complexity Metrics. Technical report 2001CRD176, GE Research and Development Center, (2001)
6. Wang, H., Zhang, D., and Shin, K.G.: Detecting SYN Flooding Attacks, In *Proceedings of IEEE INFOCOM – The Conference on Computer Communications*, vol. 21, no. 1, (2002) 1530-1539
7. Lee, C., Noh, S., Choi, K., and Jung, G.: Characterizing DDoS Attacks with Traffic Rate Analysis, In *Proceedings of the IADIS e-Society*, vol. 1, (2003) 81-88



8. Noh, S., Lee, C., Choi, K., and Jung, K.: Detecting Distributed Denial of Service (DDoS) Attacks through Inductive Learning, Lecture Notes in Computer Science(LNCS), Springer-Verlag, vol. 2690, (2003) 286-295
9. Seo, J., Lee, C., and Moon, J.: Defending DDoS Attacks Using Network Traffic Analysis and Probabilistic Packet Drop, In Proceedings of the Third International Conference on Grid and Cooperative Computing, (2004) 390-397
10. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines, Cambridge University (2000)
11. Li, M., and Vitanyi, P.: An Introduction to Kolmogorov Complexity and Its Applications, Springer-Verlag, Section 7.6, (1997) 506-509
12. Ruping S.: mySVM – a Support Vector Machine, University of Dortmund (2004)
13. Burges. C.: LA Tutorial on Support Vector Machines for Patter Recognition, Data Mining and Knowledge Discovery, Boston, 1588

# PPF Model with CTNT to Defend Web Server from DDoS Attack\*

Jungtaek Seo<sup>1</sup>, Cheolho Lee<sup>1</sup>, Jungtae Kim<sup>2</sup>,  
Taeshik Shon<sup>3</sup>, and Jongsub Moon<sup>3</sup>

<sup>1</sup> National Security Research Institute,  
KT 463-1, Jeonmin-dong, Yuseong-gu, Daejeon 305-811, Republic of Korea  
{seojt, chlee}@etri.re.kr

<sup>2</sup> Graduate School of Information and Communication, Ajou University, Republic of Korea  
coolpeace@ajou.ac.kr

<sup>3</sup> CIST, KOREA University, 1-Ga, Anam-dong, Sungbuk-Gu, Seoul, Republic of Korea  
{743zh2k, jsmoon}@korea.ac.kr

**Abstract.** We present a probabilistic packet filtering (PPF) model to defend the Web server against Distributed Denial-of-Service (DDoS) attacks. To distinguish abnormal traffics from normal ones, we used Concentration Tendency of Network Traffic (CTNT). The CTNT mechanism computes the ratio of a specific type of packets among the total amount of network packet, and detects abnormal traffic if and only if the computed ratio exceeds the ratio in normal situation. If the CTNT mechanism detects DDoS attacks, the proposed model probabilistically filters the packets related to these. The simulation results demonstrate it is useful to early detect DDoS attacks. Furthermore, it is effective to protect the Web servers from DDoS attacks.

## 1 Introduction

These days, many businesses are performed in an opened e-Society named as the Internet, especially Web environments. However, those kinds of environments are very vulnerable to Distributed Denial-of-Service (DDoS) attacks [1], [2]. In February 2000, several high profile sites including Yahoo, Amazon, and eBay were brought down for hours by DDoS attacks. As we can see the incident, most of Web servers are exposed to DDoS attacks.

In order to cope with the threat, there have been many researches on the defense mechanisms including the mechanisms based on real-time traffic analysis technique [3], [4], [5], [6], [7], [8]. However, the previous mechanisms have some drawbacks such as overhead for managing IP address and lack of commonness. In this paper, we discuss these shortcomings of previous works in detail and propose Probabilistic Packet Filtering (PPF) model to solve the flaws.

The proposed model distinguishes abnormal traffics from normal ones based on Concentration Tendency of Network Traffic (CTNT). The CTNT monitors the ratio

---

\* This work was supported by the Ministry of Information Communication, Korea, under the Information Technology Research Center Support Program supervised by the IITA.

of a specific type of packets among the total amount of network packet. The ratio is divided into TCP flag rate and Protocol rate. The TCP flag rate means the ratio of the number of a specific TCP flag to the total number of TCP packets. The protocol rate means the ratio of specific protocol (e.g. TCP, UDP, and ICMP) packets to total amount of IP protocol packets. If the proposed model detects DDoS attack using the CTNT mechanism, it probabilistically filters suspicious packets to protect the Web server against the DDoS attacks. Performance simulation of the proposed model on synthetic topologies shows that the proposed model is useful to early detect DDoS attacks and it is effective to protect Web servers against DDoS attacks

This paper is organized as follows. In section 2, we analyze other researches to detect and defend DDoS attacks. Section 3 shows the differences between Web service traffic and DDoS attack traffic at the point of CTNT's view. This is followed by the detailed describing the proposed model in section 4. The experimental results of filtering suspected packets are shown in section 5. We summarize our research and mention future work in section 6.

## 2 Analysis on the Previous Works

An efficient management of network traffic helps reducing the damage caused by DDoS attacks. Accordingly, a lot of current researches are focusing on managing network traffic to defend DDoS attacks [5], [7]. Kargl. divides network bandwidth into several queues which have different network bandwidth using Class Based Queuing (CBQ) techniques, then classify network packets and make them flow through the classified queue in each [5]. For instance, if normal network traffic flows through a high bandwidth queue and DDoS attack traffic flows through a queue of low bandwidth, flooding packets of the DDoS attacks can be reduced. However, this defending scheme needs IP address management because classifying packet is done by watching the IP address. Thus, this defending scheme needs unreasonable overhead. Ricciuli. randomly drops a SYN flooding packet to insert a new SYN packet [7]. However, this method is useful to defend only SYN flooding attacks. Table 1 shows the analysis of related works.

**Table 1.** Analysis of related work

	<b>Kargl</b>	<b>Ricciuli</b>	<b>Gil and Poletto</b>	<b>Wang</b>	<b>Kulkarni</b>
Detection	Spoofted IP addresses	Heuristic	Disproportion between "from-rate" and "to-rate"	Difference between SYN and FIN in TCP traffic	Kolmogorov complexity metrics
Defending	CBQ	Random Drop	Not supported	Not supported	Not supported
Advantages	Strong defense	Simple and effective	Applicable to backbone routers	Early Detection, Applicable to any location	Detect any type of DDoS attacks
Disadvantages	Overhead for managing IP addresses	Only for SYN flooding attacks	Only for non-spoofed IP addresses	Only for SYN flooding attacks	Overhead for managing the metrics

Detecting the DDoS attacks is an essential step to defend DDoS attacks. Thus, there have been many researches to detect the DDoS attacks [4], [6], [8]. When DDoS attacks occur, there is a big mismatch between the packet flows “to-rate” toward the victim and “from-rate” from the victim. Gil and Poletto propose the method that examines the disproportion between “to-rate” and “from-rate” in order to detect DDoS attacks [3]. However, it is not applicable to detect attacks using IP spoofing. Kulkarni et al. presented DDoS detection method based on randomness of IP spoofing [6]. Many DDoS attackers use IP spoofing to hide their real IP addresses and locations. Additionally, the spoofed IP addresses are generated randomly. The characteristic of randomness may reveal the occurrence of DDoS attacks. Kulkarni’s method uses Komogorov complexity metrics to find randomness of source IP addresses in network packet headers [9]. However it does not prohibit the DDoS attacks that do not use randomly generated address. Wang et al. proposed the method that detects DDoS attack based on the protocol behavior of SYN-FIN(RST) pairs [8]. In the normal situation, the ratio of *SYN* and *FIN* is balanced because of the characteristic of the TCP 3-Way handshake. However, the ratio of SYN packet increases drastically during the SYN flooding attack. By monitoring sudden change of the ratio of *SYN* and *FIN*, the method detects SYN flooding attacks. However it is only applicable to SYN flooding attacks.

### 3 Web Service Traffic Analysis

In a normal situation, network traffic rate has specific characteristics. For instance, SYN and FIN are in the ratio of 1:1 and TCP and UDP traffic are in the ratio of 9:1. However, in an abnormal situation (e.g., SYN flooding, UDP flooding), these ratios are broken. Using this fact, the proposed model distinguishes a normal situation and abnormal situation, and drop attack packet probabilistically. In this section, we show the differences between normal web traffic and attack traffic. To analyze web traffic, we use the CTNT method that proposed in the earlier study [10], [11]. Details of the CTNT and the differences of normal traffic and attack traffic are explained in section 3.1 and 3.2.

#### 3.1 Concentration Tendency of Network Traffic

CTNT (Concentration Tendency of Network Traffic) is defined as a phenomenon that network traffics are mainly composed of one or more specific types of network packets. For instance, almost all TCP packets have ACK flags in their headers during their connection sessions. Since the Internet has dominant network services such as WWW, E-mail, FTP etc, which are dependent on specific network protocols, CTNT can be found on not only endpoint clients and servers but also core backbone networks [12].

To analyze web traffic, we use the CTNT method that proposed in the earlier study [10], [11]. It examines the occurrence rate of a specific type of packets within the stream of monitored network traffic, and computes TCP flag rate and Protocol rate. The TCP flag rate means the ratio of the number of a specific TCP flag to the total number of TCP packets. The protocol rate means the ratio of specific protocol (e.g. TCP, UDP, and ICMP) packets to total amount of IP protocol packets. TCP flag rate and protocol rate is defined in the equation (1) and (2), respectively. In the equation,

'td' is the time interval used to calculate the value. The direction of network traffic is expressed as 'i' (inbound) and 'o' (outbound).

$$R_{td}[F i | o] = \frac{\sum \text{flag}(F) \text{ in a TCP header}}{\sum \text{TCP packets}} \tag{1}$$

$$R_{td}[[TCP|UDP|ICMP]i | o] = \frac{\sum [TCP|UDP|ICMP] \text{ packets}}{\sum IP \text{ packets}} \tag{2}$$

### 3.2 Network Traffic Analysis

In this section, we analyze normal Web traffic and DDoS attack traffic using the CTNT and show differences between them. The network traffic analyzer is made using *libpcap* to capture the network traffic. The analyzer captures network traffic and calculates TCP flag rates and protocol rates in a manner of the CTNT.

#### 3.2.1 Normal Web Service Traffic

This section shows the characteristics of normal Web service traffic without any DDoS attacks. We used *SPECweb99* to generate normal web service traffic. This tool sends HTTP requests to the Web server and receives HTTP replies from the Web server like the real Web browsers do.

Fig. 1 shows the experimental results of *SPECweb99*. We changed Simultaneous Connections (SC) to 5, 10, 50, 100, and 150, and Requests per Connection (R/C) to 1, 2, 5, and 10. As a result, the experiments show that Web service traffic has a

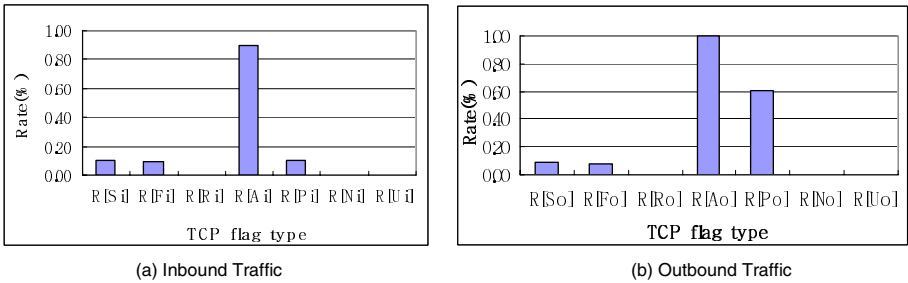


Fig. 1. Web service traffic (average value) using *SPECweb99*

Table 2. The averages and the standard deviations of occurrence rates of packets

In	R[Si]	R[Fi]	R[Ri]	R[Ai]	R[Pi]	R[Mi]	R[Ui]	R[TCPI]	R[UDPI]	R[ICMPI]
Avg.	0.17	0.00	0.16	0.67	0.16	0.00	0.00	1.00	0.00	0.00
StdDev	0.01	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00
Out	R[So]	R[Fo]	R[Ro]	R[Ao]	R[Po]	R[No]	R[Uo]	R[TCPO]	R[UDPO]	R[ICMPO]
Avg.	0.20	0.20	0.00	1.00	0.60	0.00	0.00	1.00	0.00	0.00
StdDev	0.01	0.01	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00

constant pattern with regardless of *SC* and *R/C*. The resulting rate of *SYN* and *FIN* is almost identical. The other distinguishing result is that the rate of *ACK* is very high. It's because HTTP is based on TCP which is a connection-oriented protocol. These results show that network traffic of normal Web services has a specific pattern. Table 2 shows the specific pattern of the Web service traffic.

### 3.2.2 DDoS Attack Traffic

In this section, we discuss the changes of network traffic when a Web server is attacked by various DDoS attacks. Fig. 2 shows the change of network traffic when a SYN flooding attacks occur. We generate Web service traffic during 72 seconds after 10th second from start the simulation, and a SYN flooding attack was generated during 40 seconds after 17th second from start the generation of the Web service traffic. As shown in Fig. 2-(a), the rates of *SYN* and *URG* increased to almost 1.0 and the rates of other flags, especially *ACK* rate, decreased to almost 0.0 during SYN flooding attacks.

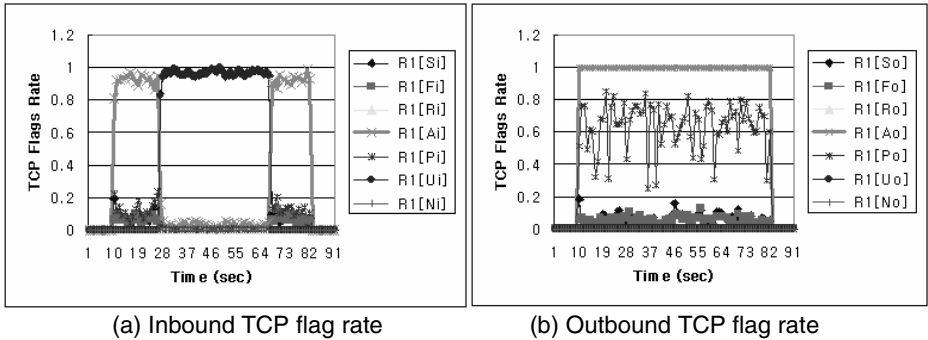


Fig. 2. SYN flooding attacks against the Web server. Under SYN flooding attacks, the rates of SYN and ACK of inbound traffic change significantly.

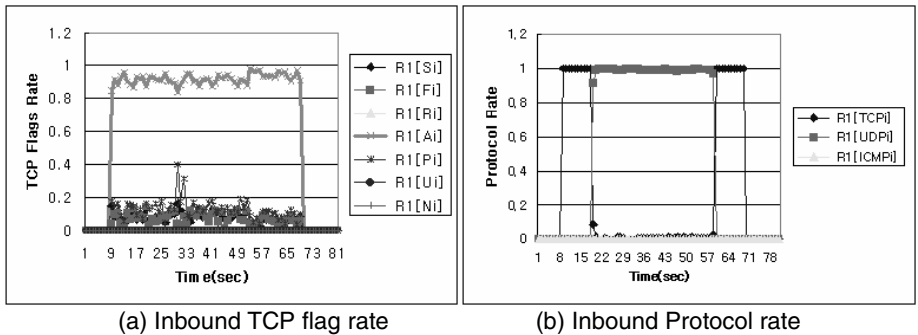


Fig. 3. UDP flooding attacks against the Web server. During UDP flooding attacks, changes are made in only inbound protocol rates.

Fig.3 presents the changes under UDP flooding attacks. UDP flooding attack occurs from 18th to the 60th second. During the attack, the rate of *UDP* drastically increases almost from 0.0 to 1.0 and TCP drastically decrease almost from 1.0 to 0.0 in Fig.3-(b). However, there is no significant change in the others.

We examined the changes of network traffic characteristics under typical DDoS attacks (SYN, UDP, ICMP flooding attacks), and found significant differences between normal Web service traffic and DDoS attack traffic as mentioned in this section. We believe that we can early detect and defend DDoS attacks by using these differences and changes of network traffic. Detail of the detection and defense mechanism are explained in section 4.

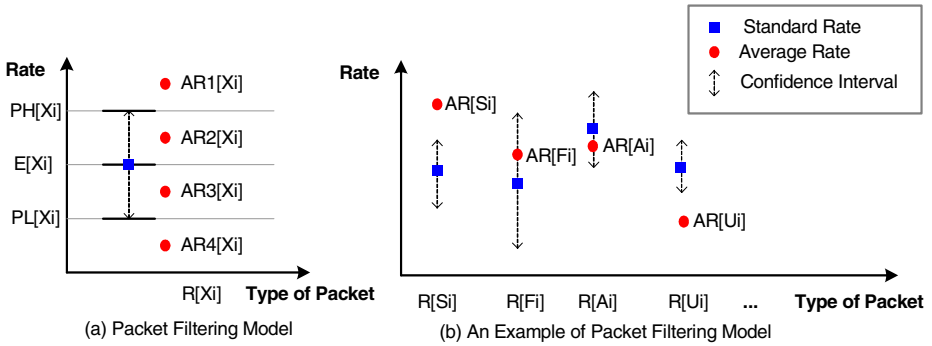
## 4 The Proposed Probabilistic Packet Filtering Model

As shown in the previous section, the rate of specific type of packet exceeds that of normal situation during an attack situation. Thus, if we always maintain the rate of normal situation, we can mitigate the effect of the DDoS attacks.

The proposed Probabilistic Packet Filtering (PPF) model is similar to the Random Early Detection (RED), which is one of active queue management models and used for the purpose of congestion avoidance on network router equipments [13], [14]. The RED doesn't drop the packets when an average queue size is smaller than *Minimum Threshold*, drops the packets with the probability varying from 0.0 to *Maximum Probability* when an average queue size is greater than *Minimum Threshold* and smaller than *Maximum Threshold*, and drops all the packets if the average queue size is greater than *Maximum Threshold* [14].

The RED algorithm behaves according to the queue size of entire packets and doesn't discriminate attack packet from normal packet. Thus, most legitimate packet is dropped with attack packet during DDoS attack. On the other hand, the proposed model acts according to the occurrence rate of a specific type of packets (i.e., TCP flag rate and Protocol rate). The rate of specific type of packet is excessively higher than that of normal situation during DDoS attacks. Thus, the proposed model effectively distinguishes attack packet from normal packet using TCP flag rate and Protocol rate, and drops attack packet without dropping of legitimated packet.

Fig. 4-(a) describes the PPF model proposed in this paper. Let the currently analyzed network traffic rate by the CTNT as Current Rate (*CR*), average traffic rate from the initial time to the current time as Average Rate (*AR*), and network traffic rate of normal traffic as Standard Rate (*SR*). In this case, the rates of normal web traffic are the values in the Table 2 of section 3.2.1. Current *AR* is calculated using an exponentially weighted average of previous *CR* values. If the previous *CR* values are non zero, current *AR* is defined by equation (3). Otherwise, current *AR* is defined by equation (4). The weight,  $w_q$ , determines how rapidly *AR* changes in response to changes in actual current rate. Floyd et al. recommend a quite small  $w_q$  to prevent the algorithm reacting to short bursts of congestion [14]. However, the proposed algorithm adopts big  $w_q$  (e.g., 0.5) since bursts of traffic are very serious threat during DDoS attack.



**Fig. 4.** Proposed PPF model; if the average occurrence rate of a type of packet  $X$  is  $E[Xi]$  in normal environment, we have confidence interval from  $P_L[Xi]$  to  $P_H[Xi]$

$$AR_{cur} = (1 - w_q) \times AR_{prev} + CR \times w_q \tag{3}$$

where  $AR_{cur}$  is Current Average Rate and  $AR_{prev}$  is Previous Average Rate

$$AR_{cur} = (1 - w_q)^m \times AR_{prev} \tag{4}$$

where  $m$  is the amount of time CTNT value was zero

In the proposed model, if average rate of a specific type of packet  $AR$  is less than lower bound of confidence interval  $P_L$  (e.g.,  $R[U_i]$  in Fig. 4-(b)), the incoming packet is serviced. On the other hand, if  $AR$  is greater than or equal to upper bound of confidence interval  $P_H$  (e.g.,  $R[S_i]$  in Fig. 4-(b)), the incoming packet is automatically discarded. Between  $P_L$  and  $P_H$  is denoted by the critical region. In this region, PPF assigns a probability of discard to an incoming packet (e.g.,  $R[F_i]$  and  $R[A_i]$  in Fig. 4-(b)). The probability depends on the factor; the closer  $AR$  to  $P_H$ , the higher the probability of discarding. The confidence interval ( $P_L$  to  $P_H$ ) and the probability of discard ( $P_d$ ) are defined by equation (5) and (6), respectively. In the equation (5), the proposed mechanism used 95% confidence level according to our preliminary test results.

$$E - 1.96 \times SD \leq R \leq E + 1.96 \times SD \tag{5}$$

$$P_L \leq R \leq P_H$$

$$P_d = \frac{AR - P_L}{P_H - P_L} \tag{6}$$

In the Table 2, for example, the average and the standard deviation of  $R[S_i]$  are 0.17 and 0.01, respectively. Then, we have confidence interval from 0.15(= $P_L[S_i]$ ) to 0.19(= $P_H[S_i]$ ) at a 95% of confidence level. If we assuming that  $AR1[S_i]$ (0.40),  $AR2[S_i]$ (0.18),  $AR3[S_i]$  (0.16), and  $AR4[S_i]$ (0.10),  $AR1[S_i]$  must be dropped because it exceeds  $P_H[S_i]$ , and  $AR4[S_i]$  should be accepted because it is lower than  $P_L[S_i]$ . On the other hand,  $AR2[S_i]$  and  $AR3[S_i]$  may be dropped or accepted according to the calculated probabilities 75% and 25%, respectively. Thus, as  $AR$  is close to  $P_H$ , the more packets are discarded.



## 5 Experimental Results

In order to evaluate the effectiveness of the proposed model, we construct simulation network and build attack model against the Web server using DDoS attack tools such as *TFN2K*. Detail of experimental environments and results are explained in the next two sections.

### 5.1 Experimental Environment

Fig.5 shows the network configuration to evaluate our DDoS defending mechanism in a simulated environment. The locations of web clients and DDoS attackers are randomly selected.

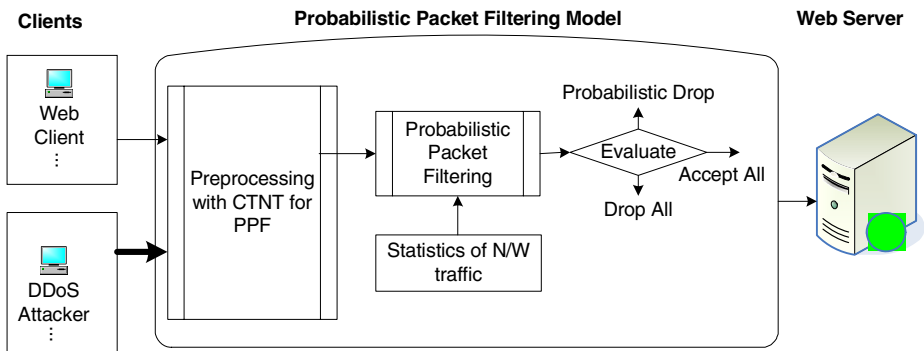


Fig. 5. Experimental Environment

Web clients send HTTP requests to and receive HTTP documents from the Web server using *SPECweb99*. While the normal Web traffic flows between Web clients and Web server, DDoS attackers generating flooding traffic against the Web server using *TFN2K*. *TFN2K* has all characteristics of other DDoS tools. We used Linux based *Apache* for the Web server.

The DDoS protector captures the network traffic both inbound and outbound one, analyzes them using the CTNT, determines drop probability of each packet, and finally forwards or drops the network packets. It works on the Linux 2.4.18 and uses *libpcap* to capture the network traffic and raw socket to forward the packets.

### 5.2 Experimental Results

Table 3 shows the experimental results of the proposed DDoS defense model. The normal Web service traffic flows during 60 seconds and the attacks using *TFN2K* are done between 20th second and 40th second.

As we can see in Table 3, the proposed defense mechanism shows very high performance in defending DDoS attacks. In the experiment, most of DDoS attack packets are dropped by PPF model with extremely low false positives; in most of attack cases the false positive rate is zero except for the case of SYN flooding attack.

During the DDoS attacks, the  $AR$  values excessively exceed the traffic rate of the normal situation as shown in section 3. Moreover, in the normal situation, UDP packet rate and ICMP packet rate are almost zero. It means that the normal web traffic is scarcely dropped since it rarely contains these packets. On the other hand, in SYN flooding attacks, there is 0.57% false-positive rate since some legitimated SYN packets are generated while average  $R[S_i]$  is higher than standard  $R[S_i]$ . Nevertheless, almost all the attacking packets are dropped by our defending mechanism.

**Table 3.** Performance of our defense mechanism. Our packet dropping mechanism helps reduce the damage of DDoS attacks.

Packet Attack	Received Packets		Dropped Packets		Drop Rate (%)		Overall
	normal	attack	normal	attack	normal	attack	
No attack	9,187	0	0	0	0%	0%	100%
SYN flooding	9,028	76,698	52	74,740	0.57%	97.45%	96.87%
UDP flooding	8,302	142,436	0	142,436	0%	100%	100%
ICMP flooding	8,545	63,674	0	63,674	0%	100%	100%

## 6 Conclusion and the Future Work

In this paper, we analyze Web traffic using CTNT mechanism and propose the Probabilistic Packet Filtering (PPF) model to protect Web servers from DDoS attacks. Our PPF model has not only an idea of RED mechanism to Internet traffic control, but also a mechanism to drop invalid packets based on 95% confidence level in accordance with an appropriate thresh hold. Our proposed model filters the suspected packets after detecting DDoS attacks via CTNT presented in the earlier study. Experimental results show very excellent results. Most of attacking packets are blocked by the proposed defending mechanism. Overall performances of our mechanism were 96.87%, 100%, and 100% on SYN, UDP, and ICMP flooding attack in each. Moreover, false-positive rate was only 0.57%. Therefore, we think our reasonable experiment results would be useful in Internet environments to defeat DDoS attacks.

In future work, we will try to evaluate our proposed model in more various situations and apply our proposed model for other specific targets such as a variety of application servers and Internet worms, especially high speed propagating worms.

## References

1. Garber, L.: Denial-of-Service Attacks Rip the Internet, IEEE Computer, vol. 33(4), (2000) 12-17.
2. Houle, J.K., and Weaver, M.G.: Trends in Denial of Service Attack Technology, CERT Coordination Center, (2001).

3. Gil, T.M., and Poletto, M.: MULTOPS: a data-structure for bandwidth attack detection, In Proceedings of the 10th USENIX Security Symposium, (2001) 23-38.
4. Householder, A., Manion, A., Pesante, L., and Weaver, M.G.: Managing the Threat of Denial-of-Service Attacks, CERT Coordination Center, (2001).
5. Kargl, F., Maier, J., and Weber, M.: Protecting Web Servers from Distributed Denial of Service Attacks, In Proceedings of the 10th International Conference on World Wide Web, (2001) 514-524.
6. Kulkarni, A.B., Bush, S.F., and Evans, S.C.: Detecting Distributed Denial-of-Service Attacks Using Kolmogorov Complexity Metrics. Technical report 2001CRD176, GE Research and Development Center, (2001).
7. Riccioli, L., Lincoln, P., and Kakkar, P.: TCP SYN Flooding Defense, Communication Networks and Distributed Systems Modeling and Simulation, (2000).
8. Wang, H., Zhang, D., and Shin, K.G.: Detecting SYN Flooding Attacks, In Proceedings of IEEE INFOCOM – The Conference on Computer Communications, vol. 21, no. 1, (2002) 1530-1539.
9. Li, M., and Vitanyi, P.: An Introduction to Kolmogorov Complexity and Its Applications, Springer-Verlag, Section 7.6, (1997) 506-509.
10. Lee, C., Choi, K., Jung, G., and Noh, S.: Characterizing DDoS Attacks with Traffic Rate Analysis, In Proceedings of IADIS International Conference on e-Society 2003, vol. 1, (2003) 81-88.
11. Seo, J., Lee, C., and Moon, J.: Defending DDoS Attacks Using Network Traffic Analysis and Probabilistic Packet Drop, In Proceedings of the Third International Conference on Grid and Cooperative Computing, (2004) 390-397.
12. Paxson, V.: Growth Trends in Wide-Area TCP Connections, IEEE Network, vol. 8, (1994) 8-17.
13. Braden, B., et al.: Recommendations on Queue Management and Congestion Avoidance in the Internet, RFC 2309, (1998).
14. Floyd, S., and Jacobson, V.: Random Early Detection (RED) gateway for Congestion Avoidance, IEEE/ACM Transactions on Networking, vol. 1, no. 4, (1993) 397-413.

# Efficient Key Agreement for Merging Clusters in Ad-Hoc Networking Environments\*

Sooyeon Shin and Taekyoung Kwon

Information Security and Computer Networks Lab.,  
Sejong University, Seoul 143-747, Korea  
shinsy80@sju.ac.kr, tkwon@sejong.ac.kr

**Abstract.** In this paper, we study a simple scheme that can effectively deal with merging different adjacent clusters in ad-hoc networks. When nodes of each cluster have already agreed on their own group keys and intend to merge themselves for further secure communications, our scheme can be used in an efficient and secure way.

## 1 Introduction

Ad-hoc networks consist of mobile nodes without any underlying infrastructure and are also referred to as MANETs (Mobile Ad-hoc NETWORKS) [4]. Each node should perform a function of router to transmit data to each other in the absence of infrastructure. Additionally, mobile nodes are limited by a range of radio coverage and have an irregular source of power because of using a battery. MANET Working group in IETF (Internet Engineering Task Force) works for standardization of such ad-hoc networks and mainly decides standards of routing protocols. In ad-hoc networks, there could be more than one group of nodes, while the group is occasionally regarded as a cluster.

**Contributions of Our Scheme.** In this paper, we propose a simple key agreement scheme for merging clusters. We consider that two different groups intend to merge themselves under their respective group keys. Our study can trivially be extended to  $n$  groups. In other words, two groups having different shared secrets such as passwords (e.g.  $PW_A$  and  $PW_B$ ) and group keys (e.g.  $K_A$  and  $K_B$ ) want to communicate together by merging clusters. For the purpose, two groups should agree on a new group key or new session key because they have different keys at merging themselves. In general, two ways can be considered for this; 1) to agree on a new group key, and 2) to reuse the established group key which is used before merging. The first may be expensive with regard to computation and communication costs depending on a group key agreement protocol. For example, performing arithmetic operations for group key agreement will be increased exponentially according to the increasing number of nodes. The second is more

---

\* This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment).

efficient than the first one but special care must be taken with respect to security concerns. Therefore, we focus on the second way in order for an efficient method of group key agreement for merging two different groups in ad-hoc networks. We divide our protocol into two parts; GKA (Group Key Agreement) protocol in each cluster and MCP (Merging Cluster Protocol) between two clusters.

The rest of this paper is organized as follows. In Section 2, we describe two essential tools for our protocol; the basic and essential routing protocol, and Group Diffie Hellman Key Agreement. In Section 3, we describes the overview of GKA protocol and MCP and in Section 4, we describes analysis of the proposal protocol. In Section 5, we present our conclusions.

## 2 Related Work

We need two basic tools for designing our protocol; ad-hoc routing protocols, and group Diffie-Hellman key exchange. Ad-hoc routing protocols are basic and necessary for our protocol since our protocol should be executed in ad-hoc networks. Moreover, the group Diffie-Hellman is suitable for group key agreement. Especially, we use the Bresson's scheme et al. [3] because it is provably secure in the random oracle model. In the following subsections, we describe these basic tools.

### 2.1 Ad-Hoc Routing Protocols

There are three different types of routing protocols in ad-hoc networks; table-based routing protocol, on-demand routing protocol and hybrid routing protocol. Proactive or Table-based routing protocol always updates and keeps tables, storing all of the path setting information from nodes to nodes by performing a constant path search work. DSDV (Destination Sequenced Distance Vector) [11] and OLSR (Optimized Link State Routing) [8] are typical examples of Table-based routing protocol. Reactive or On-Demand routing protocol updates path setting information when there is a request from a start node. The examples of On-Demand routing protocol are DSR (Dynamic Source Routing) [10] and AODV (Ad-hoc On-demand Distance Vector routing) [12]. Last but not least, Hybrid routing protocol combines the advantages of both table-based and on-demand routing protocol. For example, there are ZRP (Zone Routing Protocol) [6] and CBRP (Cluster-Based Routing Protocol) [9]. Moreover, various protocols are provided and studied for improving performance of routing protocols. There is no guarantee that a path between two nodes would be free of malicious nodes because of the absence of infrastructure and the consequent absence of authorization facilities. Thus, messages transmitted over such path can come under various attacks such as eavesdropping, altering, impersonation and routing disruption, and secure routing is necessary. Most above routing protocols do not have considered security despite that ad-hoc networks are more susceptible to routing attacks. Secure routing has been studied in various research. The representative ones are SAR (Security-Aware Routing protocol) [13] and Ariadne [7]. According to applications, secure routing protocols can be applied to our scheme.

## 2.2 Group Diffie-Hellman Key Agreement

In this section, we describe an essential tool, a Diffie-Hellman protocol. We use an extended Diffie-Hellman protocol for GKA in each cluster. This extended version is a password-based group Diffie-Hellman key exchange protocol presented by Bresson et al. [3].

**Group Diffie-Hellman.** Diffie and Hellman provided the first practical scheme of public-key cryptography in 1976. The scheme provided a method whereby two principals communicating over an insecure network can agree on a secret key in [5]. The basic concept of Diffie-Hellman protocol is that two principals pick at random values  $x_1, x_2$  and exchange the values  $g^{x_1}, g^{x_2}$  in a finite cyclic group over a network, and each principal computes Diffie-Hellman secret value  $g^{x_1 x_2}$  using  $g^{x_2}$  (respectively,  $g^{x_1}$ ) received from the other principal.

**A Password-Based Group Diffie-Hellman Key Exchange.** Several 2-party Diffie-Hellman key exchange protocols are aimed to distribute a session key among two principals when the principals share a password. Bellare et al. [1] presented a formal model for this problem. Then, Bresson et al. extended the famous EKE (Encrypted Key Exchange) to multi-party setting and proved its security [3]. This scheme will be referred to as the Bresson's scheme in the rest of this paper. We use the Bresson's scheme for Group Key Agreement (GKA) protocol because it is a provably secure password-based group Diffie-Hellman key exchange protocol. The followings are summaries of bases and assumptions of EKE protocol in the Bresson's scheme. In the Bresson's scheme, security parameters,  $l_1$  and  $l_2$  are defined and the arithmetic assumed is in a finite cyclic group  $\mathbb{G} = \langle g \rangle$  of order a  $l_1$ -bit prime number  $q$ . We then use a hash function  $\mathcal{H}$  from  $\{0, 1\}^*$  to  $\{0, 1\}^{l_2}$  and consider several block ciphers, depending on the size of the input. For each integer  $i \geq 2$ , we define two families  $\mathcal{E}^i = \{\mathcal{E}_k^i\}$  and  $\mathcal{E}'^i = \{\mathcal{E}'_k^i\}$  where  $k \in \text{Password}$ . The inverse of  $\mathcal{E}_k^i$  (respectively,  $\mathcal{E}'_k^i$ ) is denoted  $\mathcal{D}_k^i$  (respectively,  $\mathcal{D}'_k^i$ ). This Password is a small dictionary of size  $N$  and nodes of the inner cluster share a low-entropy secret  $pw$  taken from this dictionary. Such encryption schemes can be instantiated with CBC mode so that each part of the plaintext depends on the entire ciphertext. Operators  $\Phi$  and  $\Phi'$  hide away exponent parts and are used for more secure group key exchange. They are defined formally in [3].

## 3 Merging Clusters Using Group Key Agreement in Ad-Hoc Networks

Mobile nodes are divided into several duplicated or separated clusters according to certain radio coverage for making ad-hoc groups from various nodes in ad-hoc networks. Figure 1(a) shows how nodes of each cluster in ad-hoc networks are divided into two groups, cluster A and B. We postulate the mobile nodes in each cluster share the same password pre-loaded from the cluster-head. There could be several ways for loading the shared password to each cluster under the control

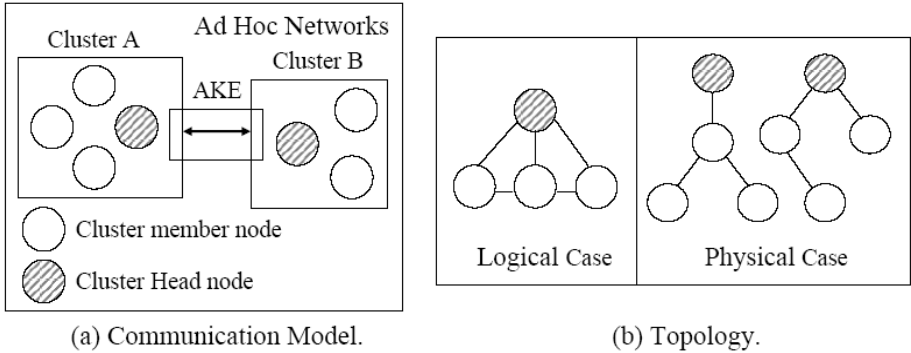


Fig. 1. Basic Concept

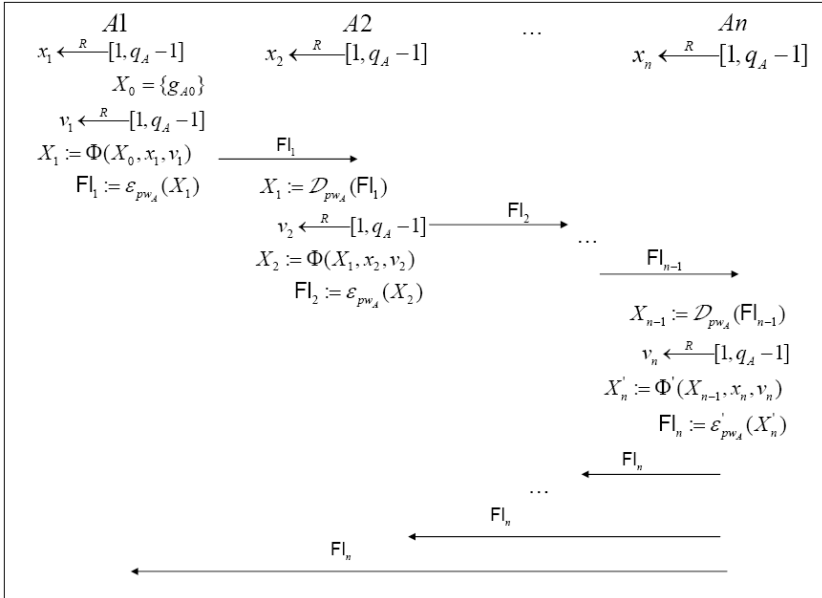
of the cluster-head or possible authority. This issue is out of scope in this paper. We assume that nodes of each cluster have already agreed on their own group key through GKA protocol and thus, two different group keys are maintained in ad-hoc networks. These group keys will be referred to as the established group key in the rest of paper. In this model, finally we assume two clusters intend to merge themselves.

### 3.1 Assumptions

**Topology of the Inner Cluster.** We divide a connection between nodes into a logical case and physical case since nodes have various topologies in ad-hoc networks. Figure 1(b) shows a logical case and physical case of topology. In a physical case, we assume that values and messages are transmitted via an adjacent different node (a neighbor node) during generating the inner group key and transmitting messages.

**Assumptions for GKA Protocol.** Nodes share a password  $PW_A$  (respectively,  $PW_B$ ) for cluster A (respectively, cluster B). We also assume that a cluster-head is a victim node since it consumes more power than others. It is possible to play a role of cluster-head among nodes by turns or voluntarily since the nodes trust each other in this stage, but we do not focus on this issue. We rather assume that an adjacent node, the closest node to a different cluster, becomes a cluster-head. Additionally, we assume that two ad-hoc groups use the same hash function, such as SHA 1 or MD 5, used to compute session keys and MAC (Message Authentication Code) for secure routing.

**Assumptions for MCP.** We define a pseudo-random function PRF. A PRF is a deterministic function  $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$  which is efficient and takes two inputs  $x, k \in \{0, 1\}^n$ . Now, we only consider  $x$  to be a variable and let  $k$  be a hidden random seed and function index,  $f(x, k) = f_k(x) = \mathcal{F}$ . We assume that it is difficult to distinguish  $g^{\mathcal{F}(x)}$  from  $g^x$  in a cyclic group of prime order  $q$  under



**Fig. 2.** GKA Protocol of Cluster A: It is based on the Bresson’s scheme et al

randomness of  $\mathcal{F}$ . In order for merging clusters, some authentication methods may be necessary between two adjacent cluster-heads for AKE (Authentication Key Exchange) in Figure 1(a). There can be at least three methods. First, they can use PKI if they are connected with an external gateway. Secondly, they can share the secret key through physical meeting each other. Thirdly, it is possible for them to be located in the same radio coverage(e.g. Cluster C). Thus, they can share the same password.

### 3.2 Group Key Agreement Protocol (GKA) in Each Cluster

In this paper, GKA protocol will perform more than twice; for cluster A, cluster B and merging clusters.

**GKA of Cluster A.** Figure 2 describes GKA protocol of cluster A and the flows ( $Fl$ ) are encrypted under password  $PW_A$ . Nodes are  $\mathcal{U} = \{A1, \dots, An\}$ . The session key space SK associated to this protocol is equipped with a uniform distribution. This protocol divides into two stages; Flow 1 ( $\rightarrow$ ) is encrypted by  $\mathcal{E}$  and decrypted by  $\mathcal{D}$ , and Flow 2 ( $\leftarrow$ ) is encrypted by  $\mathcal{E}'$  and decrypted by  $\mathcal{D}'$ .

1. Flow 1.
  - (a) We assume that a node, which wants to agree on a group key at the beginning, is  $A1$  without considering topology of nodes in ad-hoc networks. We then assume that a node  $An$ , that receives the flow finally, is



- a cluster-head.  $A_1$  just receives  $Fl_0$  = "start", and builds  $X_o = \{g_{A_0}\}$ , where  $g_0$  is a random element in  $\mathbb{G}$
- (b) A node  $A_i$  ( $1 \leq i \leq n$ ) decrypts ciphertext  $Fl_{i-1} \in \bar{\mathbb{G}}$  received from a previous node using  $\mathcal{D}_{PW_A}$  and puts it into the plaintext  $X_{i-1} \in \bar{\mathbb{G}}^i$ .
  - (c) Then, a node  $A_i$  picks at random two (private) values  $(x_i, \nu_i)$  in  $\mathbb{Z}_q^*$  and gets  $X_i := \Phi(X_{i-1}, x_i, \nu_i) \in \bar{\mathbb{G}}^{i+1}$  from the plaintext  $X_{i-1}$  according to the operator  $\Phi$ .
  - (d) Finally, a node  $A_i$  encrypts the value  $X_i$  using  $\mathcal{E}_{PW_A}$  and transmits ciphertext  $Fl_i$  to the next node in cluster A.
2. Flow 2: It starts when the last node  $A_n$  (a cluster-head) receive the last flow  $Fl_{n-1} \in \bar{\mathbb{G}}^n$ .
- (a) The node  $A_n$  decrypts the last flow received from a previous node  $A(n-1)$  using  $\mathcal{D}_{PW_A}$  and put it into the plaintext  $X_{n-1} \in \bar{\mathbb{G}}^n$ .
  - (b) The node  $A_n$  then picks at random two (private) values  $(x_n, \nu_n)$  in  $\mathbb{Z}_q^*$  and gets  $X'_n := \Phi'(X_{n-1}, x_n, \nu_n) \in \bar{\mathbb{G}}^n$  from the plaintext  $X_{n-1}$  according to the operator  $\Phi'$ .
  - (c) Finally, the node  $A_n$  encrypts value  $X'_n$  using  $\mathcal{E}'_{PW_A}$  and broadcasts the ciphertext  $Fl_n$ .

After receiving the flow  $Fl_n$ , each node decrypts  $Fl_n$  received from  $A_n$  (a cluster-head) using  $\mathcal{D}'_{PW_A}$  and put it into the plaintext  $X'_n : X'_n := \mathcal{D}'_{PW_A}(Fl_n) = \{\alpha_1, \dots, \alpha_n\}$ . Finally, each node can compute the group key  $K_A$  which is shared between nodes in cluster A and generates the session key  $sk_A$  using a hash function.

$$K_A = (\alpha_i)^{x_i} = g_n^{x_1 \cdots x_n} \quad (g_n = g_{A_0}^{\nu_1 \cdots \nu_n}) \quad (1)$$

$$sk_A = \mathcal{H}(\mathcal{U} || Fl_n || K_A) \quad (2)$$

**GKA of Cluster B.** GKA protocol of cluster B is the same as GKA protocol of cluster A except that a generator  $g$  is different ( $g_{B_0} \neq g_{A_0}$ ) since a group  $\mathbb{G}$ , which is used by nodes of cluster B, is different. Also, nodes of cluster B share a different password from a password of cluster A. ( $PW_B \neq PW_A$ ) Conclusively, cluster B shares a different group key ( $K_B$ ) from cluster A. Also, cluster B can generate the session key  $sk_B$  using  $K_B$  and a hash function.

### 3.3 Merging Clusters Between Cluster A and B

If the Bresson's scheme [3] is used for merging clusters, performing arithmetic operations will be increased exponentially according to the increasing number of nodes. Therefore, we provide an efficient way to merge clusters in this section.

**MCP.** In Section 3.1, we assumed that AKE is necessary. Thus, all messages transmitted in step 2, 3, and 4 are secure because of AKE and secure routing. We also assume that all values transmitted in cluster A or cluster B, are secure since they are encrypted by the established group key as the new session key.

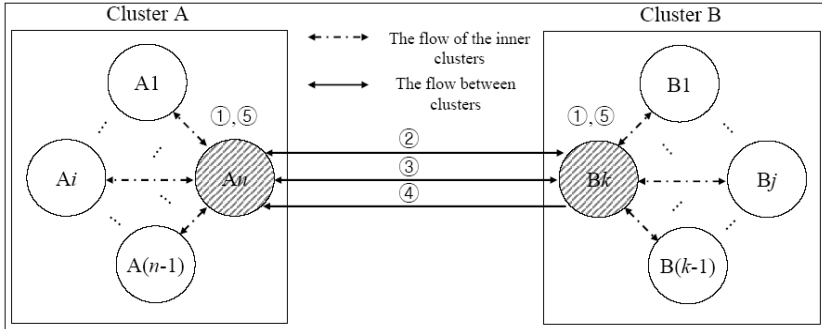


Fig. 3. Protocol for merging clusters

Figure 3 shows flows between cluster A and cluster B for MCP. We assume that the number of nodes in cluster A is  $n$  and the number of nodes in cluster B is  $k$ . Both  $n \neq k$  and  $n = k$  are possible. Before merging clusters, mutual agreement steps of domain parameters (e.g.  $g_{A0}, p_A$  and  $q_A$ ) are needed in order to generate a new group key. A mutual agreement of domain parameters is performed by choosing majority in the next Section. They are step 1, step 2 and step 3.

1. Each cluster-head sends the number of nodes in own cluster to another cluster-head.
2. Each cluster-head randomly requests routing information to nodes in another cluster in order to determine which cluster has more nodes than another. Each cluster-head then computes the number of nodes in another cluster through routing information in step 1 and reply from nodes in another cluster and compares it with own number of nodes.
3. A cluster-head, which has more nodes than another cluster-head, sends domain parameters that was used when they generated the established group key in own cluster. ( $g_{A0}, p_A, q_A$  or  $g_{B0}, p_B, q_B$ )
4. Each cluster-head picks a random variable  $x$  (respectively, variable  $y$ ) which is variable for PRF. It then computes a pseudo-random value  $\mathcal{F}_A$  (respectively,  $\mathcal{F}_B$ ) by using the established group key  $K_A$  (respectively,  $K_B$ ) which is generated in own cluster and a random variable  $x$  (respectively, variable  $y$ ): for cluster A,  $f(x, K_A) = f_{K_A}(x) = \mathcal{F}_A$  and for cluster B,  $f(y, K_B) = f_{K_B}(y) = \mathcal{F}_B$ . They exchange  $\mathcal{F}_A$  and  $\mathcal{F}_B$ .
5. Each cluster-head broadcasts a value ( $[\mathcal{F}_A, x, g_{A0}]_{K_A}$ , or  $[\mathcal{F}_B, y, g_{B0}]_{K_B}$ ) is received from another cluster-head, to member nodes.

Finally, all nodes can compute own pseudo-random value and generate a new group key using it and another one which is received from a cluster-head.

$$K_{AB} = g^{\mathcal{F}_A \mathcal{F}_B} \quad \text{where} \quad g = g_{A0} \quad \text{or} \quad g = g_{B0}. \quad (3)$$

Two clusters can create new session using a new group key  $K_{AB}$ . Consequently, two clusters merge.

**Application of Majority.** The application of majority is in order to offer fairness by choosing majority during merging clusters. Two clusters get routing information of nodes in another cluster for getting the number of nodes in the other cluster. In step 1 and 2 of Figure 3, majority is determined. Since our protocol depends on secure routing protocols, both cluster-heads can get routing information and know majority. Except that all nodes in another cluster try to deceive the number of nodes in own cluster, it is impossible for cluster-head to cheat another cluster-head because of secure routing which provides integrity. We note that the number of nodes in both clusters are the same. We assume that majority is determined by coin flipping in this case.

## 4 Analysis

In this section, we discuss an analysis of our protocol. In Section 4.1, we present security requirements and how our protocol can provide security. In Section 4.2, we describe efficiency of our protocol.

### 4.1 Security

Ad-hoc networks are more vulnerable than typical networks because of some reasons. Attackers can easily intercept messages transmitted over insecure channels since there is no fixed infrastructure and mobile nodes are wireless. Since these nodes use a power of low battery, they can not use strong cryptographic techniques. Therefore, security is important in ad-hoc networks and also for our protocol. We describe essential security requirements; confidentiality, authentication, integrity and forward secrecy and how our protocol provide these security in the followings.

**Confidentiality.** Confidentiality is the process of keeping the information sent unreadable to unauthorized nodes. To protect information transmitted over insecure channels in ad-hoc networks is necessary since information is more available not only to its intended nodes but also to eavesdroppers. One way to reach confidentiality is the use of cryptographic techniques. Encryption is a simple cryptographic technique in order to provide confidentiality. In both GKA protocol and MCP, all messages are encrypted by keys. Therefore, our protocol assures confidentiality. Semantic security, which guarantees that an adversary can not distinguish a secret value from random value. Our protocol also provides semantic security since a basic scheme of key agreement is based on Group Diffie-Hellman assumptions which have already proved that they are secure.

**Authentication.** Authentication is the process of verifying messages that are generated and transmitted from whether the claimed nodes and authorized nodes or not. It is important for ad-hoc networks to provide authentication because it is possible for malicious node or adversary to impersonate authorized nodes or legitimate nodes. In GKA protocol, we provide authentication through secure routing. AKE of MCP also guarantees authentication.

**Integrity.** When information is transmitted over insecure channels in ad-hoc networks, there is a risk that attackers see a message and change some important

data and resend it. The integrity is the ability of the secure system to guarantee that the received message is the real one that has not been altered. Also, some form of replay attacks might threaten the integrity attribute. We provide integrity in GKA protocol through secure routing. Especially, integrity is of great importance for majority of MCP. Therefore, we provide it through both secure routing and AKE.

**Forward Secrecy.**

1. Forward secrecy for GKA protocol: We assumed that cluster A (respectively, cluster B) agreed on the established group key  $K_A$  (respectively,  $K_B$ ) before merging clusters. Therefore, a communication before merging clusters should be protected from a communication after merging clusters. We emphasize that our protocol use the established group keys ( $K_A$  or  $K_B$ ) which were used in each cluster. However, that is not just using the established group key but using pseudo-random values ( $\mathcal{F}(x)$  or  $\mathcal{F}(y)$ ) of established group key. It is possible to protect the established group key before merging clusters from objects of the other cluster by PRF. Therefore,  $K_A$  or  $K_B$  cannot be recovered from  $\mathcal{F}(x)$  or  $\mathcal{F}(y)$ .
2. Forward secrecy for MCP: Also, a new group key of both cluster A and cluster B is  $K_{AB} = g^{\mathcal{F}(x)\mathcal{F}(y)}$ . We use a Diffie-Hellman arithmetic since a pseudo-random function has large randomness. Thus,  $\mathcal{F}(x)$  and  $\mathcal{F}(y)$  cannot be recovered from a new group key  $K_{AB}$ . Therefore, a new group key can be protected from external objects.

**4.2 Efficiency**

There are two cases for getting efficiency of our protocol; Generating of a new group key under the Bresson’s scheme [3] as GKA, and MCP which uses pseudo-random values of the established group keys. We compare the frequency of exponentiations, the frequency of flow transmissions according to two cases in Table 1. For this evaluation, we assume the followings. We assume that both generators and values  $p$ . We assume that the number of nodes in cluster A is  $n$ , the number of nodes in cluster B is  $k$  and the number of nodes in cluster A is more than the number of nodes in cluster B ( $n > k$ ). Also, we use the Bresson’s scheme [3] in the case of generating a new group key between cluster A and cluster B. the Bresson’s scheme uses an encryption scheme and operators, but we consider values raising generators to random values power which are chosen by nodes. We assume that the order of nodes in Flow 1 is  $A1 \rightarrow \dots \rightarrow An \rightarrow Bk \rightarrow \dots \rightarrow B1$ . ( $An$ : cluster A-head,  $Bn$ : cluster B-head) Consequently, our protocol is more efficient than generating a new group key under the Bresson’s scheme according to Table 1.

**Table 1.** Efficiency of our protocol

	Under the Bresson’s scheme	Our protocol
exponentiation	$2(n + k)$ times	$(n + k)$ times
transmissions	$3n + 3k - 4$ times	$(n - 1) + (k - 1)$ times

## 5 Conclusion

In this paper, we have provided the group key agreement protocol for merging clusters in ad-hoc networks by not generating new group key but using the established key. Namely, we have provided the efficient group key agreement protocol in order to reduce overheads for generating new group key. In addition, we evaluated that our protocol is more efficient than generating a new group key. We have also demonstrated that our protocol provide security since we assume that secure routings and AKE.

In the future study, we will implement and simulate our protocol, and consider separating clusters. Moreover, we will implement secure routing suitable for applying our scheme to ad-hoc sensor networks.

## References

1. M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated Key Exchange Secure Against Dictionary Attacks," In B. Preneel, editor, Proc. of Eurocrypt'00, LNCS 1807, pages 139-155, 2000.
2. E. Bresson, O. Chevassut, and D. Pointcheval, "The Group Diffie-Hellman Problems," In H. Heys and K. Nyberg, editors, Proc. of SAC'2002, LNCS, 2002.
3. E. Bresson, O. Chevassut, and D. Pointcheval, "Group Diffie-hellman key exchange secure against dictionary attacks," In Y. Zheng, editor, Proc. of Asiacypt'2002, 2002. Full Version available at <http://www.di.ens.fr/users/pointche>.
4. M.S. Corson and J.P. Macker, "Mobile Ad-hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations," IETF RFC 2501, 1999.
5. W. Diffie and M. E. Hellman, "New directions in cryptography," Transactions on Information Theory, IT-22(6):644-654, 1976.
6. Zygmunt J. Haas, Marc R. Pearlman, and Prince Samar, "The Zone Routing Protocol (ZRP) for Ad Hoc Networks," draft-ietf-manet-zone-zrp-04.txt, 2002.
7. Yih-Chen Hu, A. Perrig, and D. B. Johnson, "Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks," In Proceedings of MOBICOM'02, 2002.
8. P. Jacquet, P. Muhlethaler, A. Qayyum, A. Laouiti, L. Viennot and T. Clausen "Optimized Link State Routing Protocol," Internet Draft, RFC 3636.
9. M. Jiang, J. Li, and Y. Chiang Tay, "Cluster Based Routing Protocol (CBRP) Functional Specification," draft-ietf-manet-cbrp-spec-00.txt, 1998.
10. D. Johnson, D. Maltz, Y-C. Hu and J. Jetcheva, "The Dynamic Source Routing Protocol for Mobile Ad-hoc Networks," Internet Draft, draft-ietf-manet-dsr-09.txt, 2003.
11. C. E. Perkins, P. Bhagwat, "Higly Dynamic Destination-Sequenced Distance Vector (DSDV) for Mobile Computers," Proc. of the SIGCOMM 1994 Conference on Communications Architectures, Protocols and Applications, pp 234-244, 1994.
12. C. Perkins, E.Royer and S. Das, "Ad-hoc On-demand Distance Vector (AODV) Routing," Internet Draft, draft-ietf-manet-aodv-11.txt, work in progress, 2002.
13. S. Yi, P. Naldurg, and R. Kravets, "A Security-Aware Routing Protocol for Wireless Ad Hoc Networks," The 6th World Multi-Conference on Systemics, Cybernetics and Informatics, 2002.

# An Effective Method for Location Privacy in Ubiquitous Computing

Gunhee Lee<sup>1,\*</sup>, Wonil Kim<sup>2</sup>, and Dong-kyoo Kim<sup>3</sup>

<sup>1</sup> Graduate School of Information Communication, Ajou University, Suwon, Korea  
{icezzoco, dkkim}@ajou.ac.kr

<sup>2</sup> College of Electronics and Information, Sejong University, Seoul, Korea  
Tel.: +82-2-3408-3795  
wikim@etri.re.kr

<sup>3</sup> College of Information Communications, Ajou University, Suwon, Korea  
dkkim@ajou.ac.kr

**Abstract.** In ubiquitous computing environment, every service should have the characteristic of context-awareness. Since physical location is important information to grasp a user's context, the service provider should maintain the location information of the user to provide necessary service. When a malicious user acquires the location information of any other user, it is possible that he/she deduces the context of the user. Although there have been many researches for the problem, previous methods impose complex computation and many subjects requires on the system in order to gain acceptable anonymity. In this paper, we propose an effective method that protects the location privacy in ubiquitous computing environment. This method prohibits the malicious user from hijacking the location information by diffusing the information. It also confuses the attacker with transmitting dummy messages as normal users do.

## 1 Introduction

In ubiquitous computing environment, computing resources are embodied in everywhere around the people. In street, driving car, flying airplane, office, and even body, the resources are working to provide many services for the people. Further more, these services are provided timely and appropriately according to the user's context. For example, when a person falls down on a street and hurts himself/herself, someone at the vicinity calls 911 for an ambulance in the existing environment. Whereas, in the ubiquitous computing environment, the nearest computing resource, such as camera and his/her watch that includes medical sensor, is aware of the situation, and it sends an alert to the nearest hospital. The ambulance arrives in several minutes, and doctors prepare medical treatment for the injured person. To provide this kind of service environment, the system should maintain the location information of the user. Physical location of a user is important information to grasp the context of the user [1]. For example, when a professor enters the classroom at school hour, he/she may

---

\* Corresponding author.

have a class. Thus the system provides the materials for the class. In this service environment, intercepting of the location information is a serious threat on the privacy. For instance, when a malicious third party knows that a person is in a mental hospital, he/she is able to infer that the victim has a mental problem. If he/she informs the boss of the fact, the victim will be discharged. Moreover, in the ubiquitous computing environment, it is easier to intercept the message than the wired network, and the location information is acquired without the consent of the user from the intercepting message. The attacker can easily infer the context of the victim by collecting and analyzing the victim's location information within a certain period of time [2].

To protect location information, there are many researches on location privacy in various disciplines such as telemetric, mobile computing, and ubiquitous computing [3], [4], [5]. These researches try to control the usage of the location information. However existing methods have some limitations. They need complex computation to gain acceptable anonymity. Fewer subjects than the number desired by the approach do not support appropriate anonymity. Therefore, in this paper, we propose an effective approach that protects the privacy in ubiquitous computing environment. The proposed approach employs information diffusion method that scatters the user's location information in order to confuse the attacker. In addition, the base station or the access point transmits the dummy message that looks like a real traffic but has no meaning.

This paper is organized as follows. We describe private invasions, existing approach to handle it, and its limitations in section 2. Section 3 explains the proposed information diffusion approach in detail. This is followed by the describing the system architecture of the proposed method in section 4. Section 5 concludes.

## 2 Privacy Invasion in Ubiquitous Computing

### 2.1 Information Leaking vs. Information Gathering

In location-aware application, there are two kinds of method to hijack other's location information; one is unauthorized information access, called information leaking, another is inference with observation of the user's location, called information gathering. In the former case, the malicious user acquires the victim's location information from the system. In the latter case, the attacker is able to obtain location information by observing the traffic of a network, and then he/she infers what the victim does or how the victim's status is. Moreover, in this case, since both the victim and the system do not notice that the private information was indeed leaked, it is impossible to prohibit gathering the private information.

In case of the information leaking, the existing access control methods can be used to control the unauthorized access on the location information. Each system holds own policy to control the access on the information. Every request is sent to the authority that decides whether it is allowed or not. In addition to this, the monitoring system logs all the usage of the location information. When the unauthorized access is exposed, the logged data can be used to trace the malicious user.

In the case of the information gathering, since the eavesdropping of the signal does not detected by both the sender and the receiver, the malicious user is able to collect any user's location information without notice. Moreover, the malicious user can infer the locations and the user's context from the collected set of information. It is difficult to control the observation and inference attack properly. In this paper, we focus on the information gathering problem in order to prohibit the malicious user from obtaining the location information of others.

## 2.2 Previous Approaches Against the Privacy Invasion

M. Langheinrich introduces the requirements for the privacy in the ubiquitous computing environment [6]. It allows data collectors to both announce and implement data usage policies, as well as providing data subjects with technical means to keep track of their personal information as it is stored, used, and possibly removed from the system. E. Sneekenes identifies concepts that may be useful when formulating the privacy policy [7]. This policy makes that the individual should be able to adjust the accuracy depending on parameters such as the intended use and the identity of the recipient. Urs Hengartner et al. suggest the system that controls the access on the location information in the Wireless LAN based people location system [8]. It has the hierarchy of the location system and the access control mechanism delegates the request to the lower level system. After all, the victim user's device authenticates and authorizes the requester according to the victim's policy. The policy and mechanism are based on the SPKI/SDSI (Simple Public Key Infra Structure/Simple Distributed Security Infrastructure). P3P (Platform for Privacy Preference) provides negotiation between server and client [9]. At the negotiation, when a user wants to access the information, he/she receives the server's policy and compares it with the local policy. If the received policy is acceptable, he/she accesses to the information.

Bugra Gedik et al. introduce the cloaking method that provides vague temporal and spatial information in order to conceal a user within a group of  $k$  people [10]. The system employs an anonymous server act as the mix node. It prevents a malicious observer from linking ingoing and outgoing messages at the server. Different with any other  $k$ -anonymous system [5], it uses a customizable  $k$  that is changed by the environment. F. Stajano et al. solves the location privacy problem with the pseudonym that changes randomly and frequently [11]. It classifies every area into two zones; one is application zone where a user has registered for a service, another is mix zone that is a connected spatial region of maximum size in which none of users has registered for any services. Since applications do not receive any location information but pseudonym when users are in a mix zone, the identities are mixed.

Although there are many researches on the anonymity based location privacy protection method as we mentioned above, they have still some drawbacks. Indeed, it is very hard to prohibit the malicious user from gathering the location information. In the next section, this paper will discuss on the drawbacks on the previous approaches. That will give the reason why we should propose an effective method that enhances the location privacy.



### 2.3 Drawbacks on the Previous Approaches

Existing approaches provide the reasonable anonymity service if only there are enough subscribers in the network since they employ  $k$  anonymous system. If the number of subjects is smaller than the constant  $k$ , anonymity may be corrupted by the attacker.

In addition, there is large computational complexity since it changes the pseudonym frequently for the secure anonymity service. When a few subjects are in the anonymous area, the number of pseudonym change is not large since the number of location change is small. The attacker, however, may be able to perceive the pattern of pseudonym change quickly than we thought [12]. In practice, many subjects are in an area in order to achieve the reasonable anonymity. In this case, the system suffers from the extremely complex computation.

In practice, every person has his/her own pattern of movement. For example, a person, who is a salaried worker, is almost in the office at the morning. Between twelve and fourteen o'clock, he is in the restaurant for the lunch. At 4:00 p.m., he/she is in the meeting room with his/her co-worker. Thus the attacker can easily infer where the victim's signal comes from.

In the Mix zone, the area is represented as the combine of application zone and mix zone. The various users are in an area in general. When user A's registered service  $s$  and user B's registered services are different, there are two different application zones and mix zones are constructed. In practice, the subjects are generally more than 100 in the street. Then there may be too many different zones in the same area. It makes the system will forgive providing the anonymity service.

## 3 Location Information Diffusion and Confusion

### 3.1 Diffusing the Location Information

The basic idea of the diffusion is that it divides the information and inserts each fraction of the information into the normal packet. Most of service request packet size does not mount up to maximum packet size. Every packet may have many optional fields, which are not practically used, in the TCP/IP header parts. Moreover, the practical size of the internet packet is 576 bytes. That is, every packet has many unused space. The proposed system uses some parts of them. To do so, firstly the system chooses the multiple parts among the unused space. Both client and server should know the selected part consistently, but position should be changed whenever the private information is transmitted. Thus, we use selection table that stores selected part of unused space. Fig. 1 shows that how the position is selected with the table and how the information is diffused into a packet.

The size of the selection table is  $8 \times 8$ . Each cell of the table contains a list of the unused positions in the packet. Every list includes three positions. Each position may be one of the following candidate positions; IP option field (40 bytes), TCP option field (40 bytes), and data field (1380 bytes). The proposed

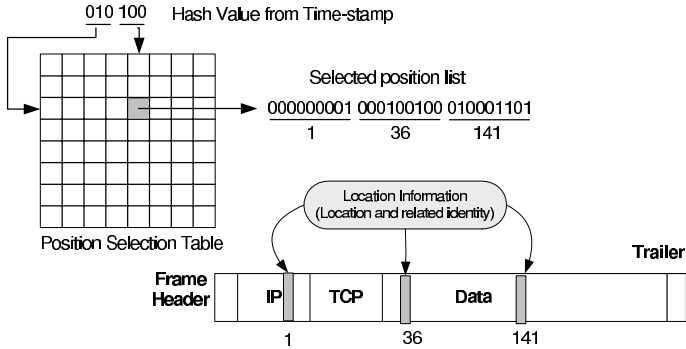


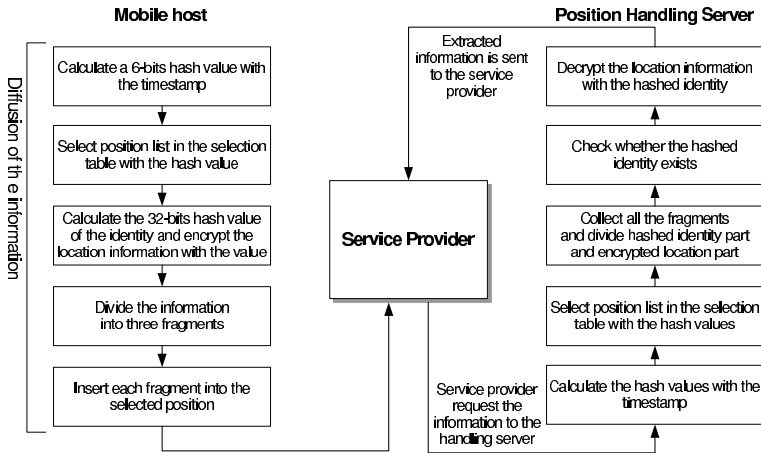
Fig. 1. The diffusion of the location and related identity information

system divides them into 365 positions of 4 bytes. Each selected position can be represented with 9 bits ( $365 < 2^9$ ). Since each cell contains 3 positions, the size of the cell is 27 bits that can be expressed as 1 word of the 32-bit machine.

Every list in the table is generated randomly by the position handling server. For a list, the server randomly selects 3 positions among 365 candidates whenever it creates the position selection table. To communicate with the server, the positions of the inserted parts are maintained consistently between the client and the server. Thus, in the proposed system, we employ the same position selection table in the both sides. The management issue will be discussed at section 3.3.

Fig. 2 shows the detailed process of the transmission of the location information. In order to select a cell having position list, the client calculates a 6-bit hash value from the time stamp. The client selects the row of the table by the first 3-bits of the hash value, and it selects the column of the table by the last 3-bits of the hash value. According to the selected list, the fractions of the location and the related identity information are inserted into a normal packet. The identity information is inserted into the packet as the form of hashed value, and the location information is inserted as the form of ciphertext that enciphered with the hashed value of the identity. The other empty positions are filled with random numbers.

When the service provider receives the packet from the client, it just transmits the packet to the position handling server. The position handling server extracts the location information, and it transmits the information to the service provider. The extraction process will be performed in the opposite order of the diffusion. The position handling server generates 6-bit hash value from the timestamp in the packet header, and then the position handling server selects a cell of the position selection table. According to the position list in the cell, the server gathers all the fragments of the location and identity information. The server checks whether the hashed identity exists or not, and then the server decrypts the location information with the hashed identity only if the identity exists. If the position handling server succeeds to extract, then it sends the location information to the service provider. Otherwise, it discards the message and notifies the fact.



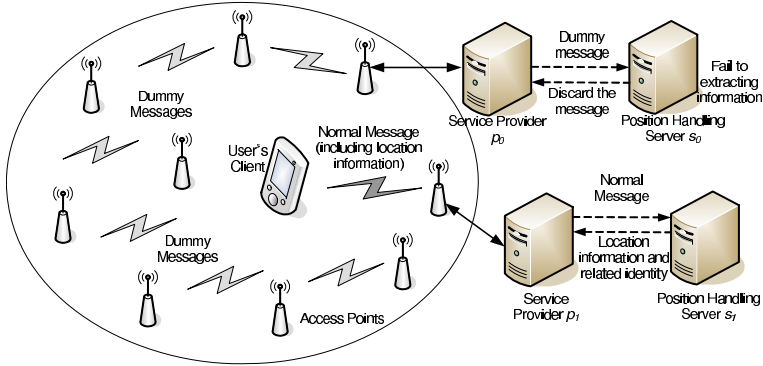
**Fig. 2.** Detailed process of the transmission of the location information

### 3.2 Confusing the Malicious User

When a user's client sends the location information, the user's traffic can be easily captured or intercepted by the malicious user in general. To confuse the malicious user, anonymity approach has been used until now. The anonymity approach, however, has high computation complexity as we previously discussed. In the proposed approach, the location and identity information are already diffused in the normal packet. Even if the attacker intercepts the packet, he/she does not know whether the packet includes the location information. In addition, he/she does not know which positions are used for the location information. Thus, instead of the complex anonymity method, we employ simple confusing method that employs the dummy user instead of the anonymity system.

The dummy user is a part of the system that acts as the normal user. The dummy user can be a base station or an access point. They periodically emit the dummy message that has no meaning but acts as normal traffic. Thus the real traffic hides behind the floods of dummy traffic. It is difficult that the attacker distinguishes the real traffic from the dummy traffic. Fig. 3 shows the schematic view of the environment that employs several dummy users. In this figure, since the dummy user and the client send the message to someone, many packets move to somewhere else. Thus the status of the network is very dizzy.

To make the dummy message, the dummy user fills the contents of the packet with several random numbers. The address may be selected randomly among several neighbor hosts in the destination list. The destination list is managed by each dummy user. When the position handling server receives a message, it is unnecessary to determine whether it is normal message or not. Since the position handling server fails to extract the location and related identity information with the dummy message filled with random values, the dummy message will be discarded at the server. As shown in Fig. 3, the position handling server  $s_0$  is unsuccessful to extract the location information since the received message is a



**Fig. 3.** Dummy users on the networks; the confused area filled with numerous dummy messages. Dummy message is discarded at the server, while normal message is properly used for providing location information.

dummy message, while the position handling server s1 succeeds in abstraction of the location information. The period of the emitting the dummy message is randomly changed.

### 3.3 The Position Selection Table Management

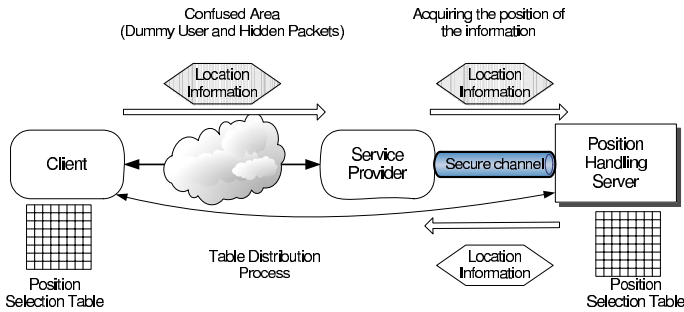
If the table is always the same, the malicious user can infer the selected position with the brute force analysis since the rate that selects the same position may comply with the uniform distribution. Although the malicious user does not know the location in real time, he/she can find out the set of the location. They collect that information for a day or a week, and then they can infer his private status such that he has a mental disease. Therefore the position selection table should be updated within reasonable amount of time  $t_{update}$ , called *update time*. That is, the proposed system will update the selection table if and only if  $t_{used} \geq t_{update}$ , where the  $t_{used}$  is the used time of the selection table and  $t_{update}$  is the update time of the table. For the decision of the update time  $t_{update}$ , there is no deterministic method but practical one. According to the service, the required degree of strength of the location privacy is various. The higher degree is required, the lower update time is decided.

The proposed system employs two table update methods; one is pull method, another is push method. In the push method, the position handling server transfers the new position selection table when a user uses the service. The detailed steps are as follows.

1. When the position handling server extracts the location information from a request, the server checks whether the used time  $t_{used}$  is larger than the predefined update time  $t_{update}$ .
2. If  $t_{used} \geq t_{update}$ , then the position handling server sends the table update message to the service provider. The service provider relays the message to the client who sent the request.

3. The client sends the ready message to the service provider. The service provider relays the message to the position handling server.
4. The position handling server creates a new selection table, and sends the table to the client via the service provider. The table is encrypted with the temporal symmetric key. The temporal key is a hash value from the combination of the location information and the user's id.
5. The client decrypts the received selection table and uses the table at the next position selection process.

In the pull method, according to the request of a user, the position handling server creates a new selection table, and then the server transfers it to the user's client. To transfer securely, the client and the service provider are temporarily connected with the secure channel such as SSL, and the table should be encrypted by the user's password that is created by the user at the registration time.



**Fig. 4.** Schematic diagram of the proposed system; three thick white arrows mean the flow of the location information. The shaded hexagon represents the hidden location information, while white hexagon represents extracted location information.

## 4 System Architecture

Fig. 4 shows the schematic view of the proposed system. It consists of three components and two different communication environments. The client is a user's mobile device such as PDA, cellular phone, and handheld PC. The service provider is a server that supplies any service. Through the service provider, the client acquires necessary service such as location for a friend and showing the way to the destination. The position handling server authenticates and authorizes the requested user. It acts as trusted third party in the public key system. The confused area is located between the client and the service provider, and all the messages from them should be transferred through the area. It helps that the system hides the user's traffic. Both the service provider and the position handling server are connected with the secure channel such as SSL or VPN since the traffic between them contains the user's identity. The location information is diffused into a message from client to service provider. The location information is extracted by the position handling server and is sent to the service provider.

Every transaction, including the location and the related identity information, between the client and the provider is handled as following brief description. The client creates a message that notifies a user's location. It is diffused and inserted into the normal packet. Through any mobile network such as wireless LAN, CDMA, or GSM, the client sends the packet to the service provider, then the provider just passes the packet to the position handling server. In the position handling server, the location and the related identity information are extracted from the received packet, and then the information is transferred to the service provider.

## 5 Conclusion

In the ubiquitous environment, every service providing system may have the characteristic of the context-awareness. One of the most important information is the user's location information in order to aware his/her context. Thus the location information and the related identity are often sent to the service provider though the network. Since the attacker may intercept the information, sending the information without protection is very dangerous.

In this paper, we propose an effective approach to protect the location and related identity information. In the proposed approach, the sender diffuses the location information into the normal traffic and dummy user emits dummy message that confuses the malicious user for higher secure level. With this approach, the location information can be transferred to the service provider securely. In the future work, the diffusion will be applied to multiple packets, and the dummy user will act like a real user.

## References

1. Garlan, D., Siewiorek, D.P.: Project Aura: Toward Distraction-Free Pervasive Computing, *IEEE Pervasive Computing* (2002) 22-31
2. Rodden, T., Friday, A., Muller, H., Dix, A.: A Lightweight Approach to Managing Privacy in Location-Based Services, *Proceedings of Equator Annual Conference* (2002)
3. Duri, S., Gruteser, M. et al.: Framework for Security and Privacy in Automotive Telematics, *WMC02, Atlanta, Georgia USA* (2002) 25-32
4. Gruteser, M., Grunwald, D.: Enhancing Location Privacy in Wireless LAN Through Disposable Interface Identifiers: A Quantitative Analysis, *WMASH03, San Diego, California USA* (2003) 46-55
5. Gruteser, M., Grunwald, D.: Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. In *MobiSys, USENIX* (2003) 31-42
6. Langheinrich, M.: A Privacy Awareness System for Ubiquitous Computing Environments, In *Proceedings of the 4th International Conference on Ubiquitous Computing, LNCS No. 2498, Springer-Verlag* (2002) 237-245
7. Sneekenes, E.: Concepts for Personal Location Privacy Policies. In *Proceedings of the 3rd ACM Conference on Electronic Commerce* (2001) 48-57
8. Hengartner, U., Steenkiste, P.: Implementing Access Control to People Location Information, *SACMAT 2004, NewYork, USA* (2004) 11-20

9. Cranor, L., Langheinrich, M., Marchiori, M., Presler-Marshall, M., Reagle, J.: The Platform for Privacy Preferences 1.0 (P3P1.0) Specification. (2004) <http://www.w3.org/TR/P3P/>
10. Gedik, B., Liu, L.: A Customizable k-Anonymity Model for Protecting Location Privacy, CERCS Technical Reports, GIT-CERCS-04-15 (2004)
11. Beresford, R., Stajano, F.: Location Privacy in Pervasive Computing. PERVASIVE computing, IEEE CS and IEEE Communications Society (2003) 46-55
12. Beresford, A., Stajano, F.: Mix Zones: User privacy in location-aware services, Proceedings of First IEEE International Workshop on Pervasive Computing and Communication Security, A Workshop in PerCom 2004 (2004)

# Integrated Support for Location Aware Security Services in Enterprise Wireless Networks

Zhaoyu Liu<sup>1</sup>, Peeyush Sharma<sup>2</sup>, and Jian Raymond Li<sup>3</sup>

<sup>1</sup> Department of Software Information Systems,  
University of North Carolina at Charlotte, Charlotte, NC-28223  
zhliu@uncc.edu

<sup>2</sup> Department of Computer Science,  
University of North Carolina at Charlotte, Charlotte, NC-28223  
psharma3@uncc.edu

<sup>3</sup> Cisco Systems Inc., 210 W Tasman Drive, San Jose, CA 95134  
rali@cisco.com

**Abstract.** In wireless computing environments limiting network access to authorized users is paramount for the overall security of network. In addition to basic authentication framework, network access is also governed by the context in which it is being used. In this work, we address security issues based on one such context: location. Location sensitivity is increasingly becoming an integral aspect of wireless and pervasive applications. As user moves around in an ubiquitous environment, access rights and other security services provided to her need to be evaluated accordingly. For such purposes we need a security mechanism that controls the authentication and other security services based on location, in addition to basic identity information. In this paper, we present an architecture of location aware security services for enterprise wireless networks. Our implementation is integrated into the RADIUS system, an Authentication, Authorization and Accounting (AAA) framework. Performance evaluation shows that our implementation is efficient for location based security services.

## 1 Introduction

The inception of pervasive computing has allowed a single user to interact with multiple processing devices. Traditional methods used for authentication like password mechanism, do not work sufficiently for security issues in pervasive environment [15]. Moreover it is not always sufficient to authenticate a user only on basis of her identity. In some instances the context of network access assumes more importance than only identification of user [9]. There is a lot of possibilities ranging from providing just the essential security, to providing strong security where same user will have different security services at different locations.

As an example of basic security need, consider Internet services in a coffee shop. It is highly desirable that network access is available only within the shop premises as the shop-owner will prefer to provide it only to customers. On the other side, consider a large enterprise network where personnel in different departments are provided different access rights. The enterprise would ensure that



basic Internet services are provided in all departments. But information considered confidential for one particular department should not be made available to an employee coming temporarily from another department.

Location authentication alone will not help in ensuring differentiated access rights. For such rights we need a security system where both authentication and authorization are part of the security infrastructure. In this paper, we present an architecture of location aware security services for enterprise wireless networks. Our implementation is integrated into the RADIUS system, an Authentication, Authorization and Accounting (AAA) framework.

The rest of this article is organized as follows. In section 2 we outline the motivation for our work. Section 3 is divided in two parts. The first half describes the working of AAA and the implementation of RADIUS. This discussion is succeeded by our design description in section 3.2. Section 4 covers implementation and performance analysis of integrated support system for location aware security services. And then we discuss the related work in section 5. Section 6 concludes the paper and presents future work of the research.

## 2 Background and Motivation

With wireless computing, we have seen the increasing use of light devices for communication and data processing. Of late, pervasive computing has paved the way for use of smart devices that easily get assimilated in the environment. In the traditional systems access-rights are determined once the credentials submitted by the user confirm with the policies set for her.

In pervasive computing needs are different. One difference lies in the short duration for which a person interacts with a device and then moves on, to access another. The interaction with smart devices is often discrete. If the user has to present her credentials for every smart device then security will become an unwieldy process for the user. The need here is to take the security aspect higher up in hierarchy. This can be achieved by establishing trust between user and environment rather than between a single user and multiple devices in the pervasive environment. This presents need for the context based authentication and access-control. In wireless and pervasive environments one of the major contextual parameters is location. Once a user can be trusted in an environment then smart devices in that environment can be setup according to the preferences of the trusted user. A situation is illustrated below to exemplify this argument.

Consider a classroom that is setup with pervasive devices. Students can use laptops or PDAs to take notes. A webcam is deployed, checking movements to and from the classroom. Multiple smart screens can be used for the course instructions. IR beacons can be used to track the movement of people who keep IR listeners with them. The security task is to associate access rights in this active space with the instructor. This can be done in two ways. Either instructor authenticates to each device in the classroom separately or the space can be setup on the basis of her presence in the smart room; confirmed by IR beacons. Same user can have different authorization levels in different locations. Continuing with the classroom

scenario, instructor should not get the authority to show the grades of all students in the classroom using the smart screens. Though the instructor has the authority to manage grades of students in her office. In such cases access permissions get determined by the location in which devices are being used.

Furthermore, it is not sufficient to check who is getting network access but necessary to ensure at what level permissions are being granted at various locations. In this work we are presenting the criteria and framework in which location awareness in wireless and pervasive computing can be used to provide varied access rights for user. To ensure a centralized system for security, we are using Remote Authentication Dial In User Service (RADIUS) [1, 14] which is a widely used implementation of AAA.

### 3 Location-Aware Architecture

The Authentication, Authorization and Accountancy (AAA) architecture provides a centralized system where network access can be validated, controlled and monitored. In this work, we are using an AAA implementation, RADIUS, to authenticate user and authorize security services by taking location as a contextual parameter. In the next sub-section we describe basic functionality of RADIUS protocol and then present our location extension design integrated into RADIUS.

#### 3.1 RADIUS System

Remote Authentication Dial In User Service (RADIUS) is an AAA access-control protocol originally developed for dial-in services. RADIUS is supported in Virtual Private Network (VPN) and wireless access networks. The basic architecture of RADIUS is depicted in Figure 1. User seeks network access through end user devices. Client takes user's credentials and submits it to the server. RADIUS provides centralized database which contains information needed for user authentication and access-control. This database also keeps a list of configuration items for each user, detailing the type of services user is entitled to receive. The authentication information submitted by user is forwarded to the RADIUS server in form of an access request. This access request contains user's name, her password, client's identification and the port number that user is trying to access.

Upon receiving the access request, RADIUS server first validates the client. If the client does not have the shared secret with the RADIUS server then the request is silently discarded. After successful validation of client the RADIUS server checks a database to find the User-Name. If the User-Name is found in the database then RADIUS checks for User-Password and some other parameters, depending on the configuration setup. In case the User-Password or other configured parameters do not match, an Access-Reject message is sent back to the client.

#### 3.2 Location Extension into RADIUS

When RADIUS is used for wireless access, Access Point (AP) works as a RADIUS client. The AP sends the Access-Request to the RADIUS server for any user who

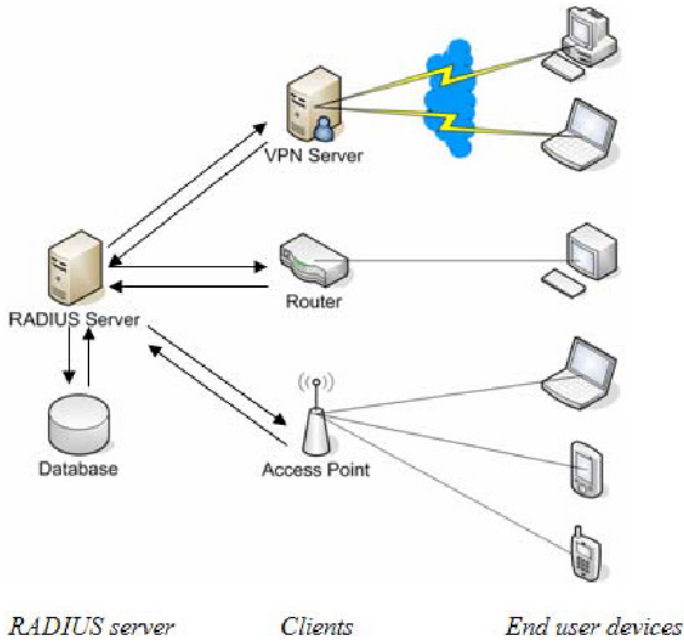


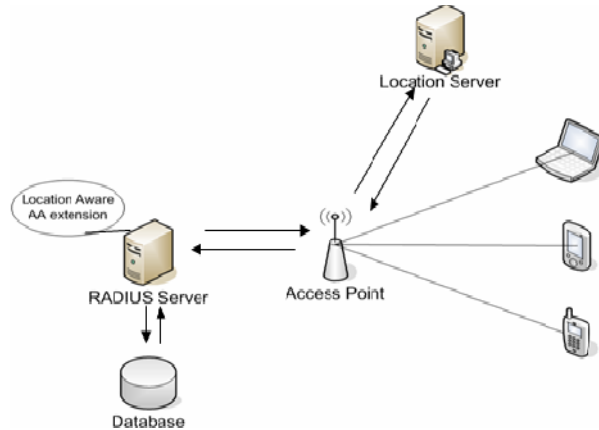
Fig. 1. RADIUS Architecture

comes under its coverage and requests access. In a network users are assigned appropriate roles according to level of information they are provided access to, and responsibilities they are accounted for [3]. There are various location aware mechanisms available which can correctly determine the location of a user. One of them is Cricket [6].

In figure 2 the location aware security services framework is shown. The Location Server is used to provide location information to the client. When AP receives the Access-Request from a host in it’s vicinity it first contacts the location server. After adding location coordinates to the Access-Request, client forwards request to server. The remote server will have at least one database setup for profile check. When the Server receives request it first authenticates user based on password mechanism set for her in the profile. Authentication is approved if the entry for user in the AA extension suggests that user is allowed to access the network in perimeter specified in the Access-Request. Policies defined in access-control manage authorization levels for users. RADIUS provides security measures for client server communication through MD5 hashing. The security mechanism will be discussed later in section 5.1.

Unified Modeling Language (UML) diagrams can be used to depict sequence of activities [2, 10]. Events taking place during location aware RADIUS authentication are shown in UML sequence diagram (Figure 3).

The communication scenario can be divided into the following steps.



**Fig. 2.** Location Enhanced RADIUS Server

1. User submits an asynchronous Access-Request to the client.
2. The client checks the user location with the location server.
3. Location server retrieves the current location information for concerned user and sends this information back to the client.
4. Client appends location entities to the original Access-Request.
5. Updated Access-Request packet including location information is forwarded to the RADIUS server.
6. The RADIUS server validates the client. Here we assume that validation succeeds.
7. The RADIUS server searches the database for users profile taking User-Name from the request as a key.
8. If user profile exists for the user then RADIUS retrieves the profile.
9. Configured attributes in user's profile are checked against attributes in the Access-Request (with location information included).
10. Upon verification of request attributes, access is granted to the end user.

In wireless environments users move within the designated wireless network area. To cover such movements security policy is maintained in a hierarchical fashion. Beacons grouped together cover a particular space within the area. User's access rights do not change within a space. But if user moves out of this space then information about access rights is transferred to the higher authority. This authority determines user's location again and grants access rights applicable for that particular location. If user completely moves out of the network area covered by a single centralized authority then credentials have to be re-submitted to the new authority, from the previous authority domain to the new authority domain. If previous and new authorities have trust established between them, then transferred user's credentials can be used to establish proper security services in the new authority domain.

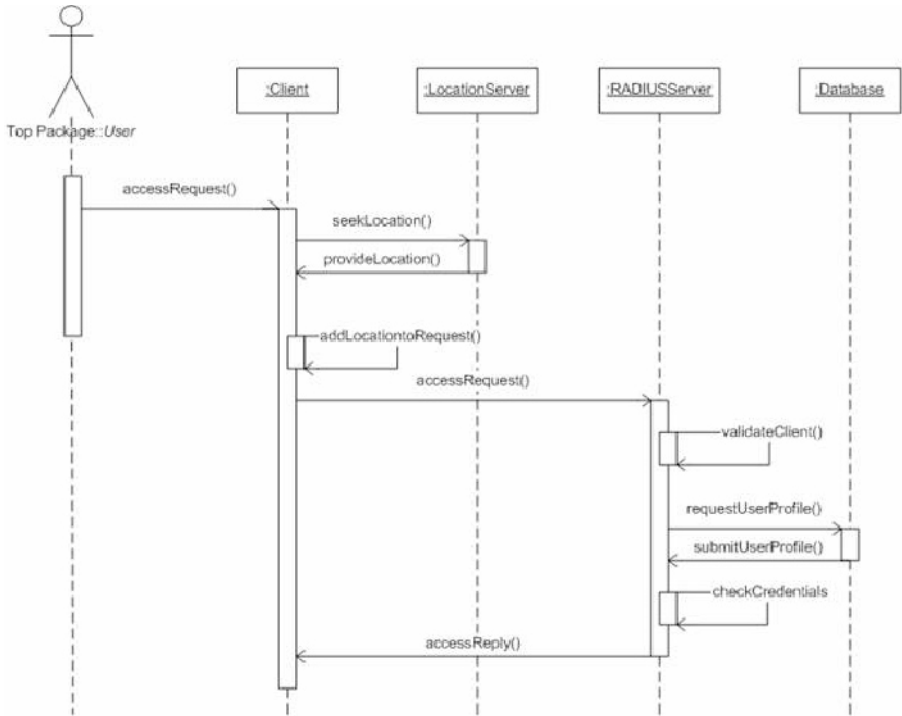


Fig. 3. Sequence Diagram for Location based Authentication

## 4 Implementation

RADIUS has been implemented by various open source software development teams and vendors. For our application we used an open source Software called FreeRADIUS version 1.0.1 on Linux platform Kernel version 2.6.11 [16].

RADIUS server side contains modules to perform configuration updates, database management and other single well-defined tasks. These modules are not needed for the basic application of RADIUS Server but they are used, to perform additional tasks. In our application a location module is added to RADIUS to manage location information. This module manages both authentication and authorization and gets activated only if the configuration for the user has been setup for location based access management. For tracking location information, environment is divided into certain sections or spaces. Location information in the Access-Request contains the space identifier for the section and coordinates for the device in that section. In the following subsections we discuss the security and performance aspects of our implementation.

### 4.1 System Security

RADIUS are vulnerable to real-time active wiretapping attacks [1]. But these attacks can be thwarted by generating unique unpredictable requests. There are two

kinds of authenticators used in RADIUS. One is request-authenticator which the client uses when sending an Access-Request and the other is response-authenticator used by the server for the Access-Response. A request-authenticator is a MD5 hash generated on a completely random basis by the client and makes the request unpredictable. Once the server receives the Access-Request and decides access for user, it calculates the response-authenticator. This response-authenticator is a MD5 hash based on packet's identification, attributes and request-authenticator. Response-authenticator is appended to the Access-Response and sent to the client. Since only the client is aware of the random request-authenticator, it can create a MD5 hash of packet's identifiers, attributes and request-authenticator. If such a hash matches with the response-authenticator received from the server then the client can be assured of the integrity of the communication. For our location enhanced service the RADIUS server includes the location information in creating response-authenticator. Upon receiving response-authenticator from server, client matches it with the MD5 hash created with location information among other identifiers. If two hashes match then the communication is considered to be secured.

## 4.2 Performance Evaluation

Experiments were conducted on the server running on Intel Pentium 4 CPU with a 3.00 GHZ speed and 1.5 GB SDRAM. The client and the location server were also running on the same system. Communication was established through network sockets. Time taken to perform a simple password based authentication was 38ms in our setup. The location enhanced authentication mechanism increased this time to 52ms. The difference in authentication time is attributed to the extra communication link introduced between the location server and the client.

## 5 Related Work

Location awareness has been identified as a key parameter for context aware applications [8, 11]. Context aware applications are identified as ones which adapt their behavior to changing environments. Determination of user's location and secured exchange of location information form the backbone of such applications.

The work in location awareness in a closed environment has been done by Priyantha et al [6]. They have designed location awareness mechanism termed Cricket for location support for in-building, mobile location dependent systems. Location awareness is achieved through beacons and listeners. Listeners are attached to a mobile device whose location is to be tracked. Beacons are small devices that are spread in the environment. Listener determines its location on the basis of signals received by various beacons in the region. In our design we use Cricket for location determination. Want et al. [12] have also designed a system for location awareness in closed office environment using active badges.

In [17] Sastry et al. discuss the secure exchange of location information. In [13] author uses address on the Internet as the location parameter. This location information has been used as off-line information to thwart attacks during multi

party communication. But this off-line information does not cover physical location of user for authentication. In [15] Bardram et al present proximity-based login which allows users to be authenticated on a device just by approaching it physically. But proximity has not been extended as a parameter for authorization. Koo et al. [14] demonstrate use of RADIUS to determine mobile device location.

## 6 Conclusion and Future Work

Location determination applications and toolkits are being deployed widely for pervasive devices in the public networks. With this trend, location awareness mechanisms will have a significant contribution to determining security services. In this paper, we propose an authentication and access control framework based on location awareness in an enterprise network, as a first step toward pervasive security. The architecture assumes that a location management system can provide users location to the system. The access rights to the user vary according to the policies set for different locations. Our approach is realized by the integration with an AAA framework, the RADIUS system. Enterprise networks with context based security policies will benefit from our system.

In the future work we plan to add support for multiple levels of security. With every authorized service the RADIUS server can determine quality of protection required for secure communication for different locations. Our location module in RADIUS works with several others modules that exist in basic RADIUS architecture. Encryption for different levels of quality of protection can be achieved with security library module working with the location module. Various applications with different security services requirements will be developed based on our system.

## Acknowledgements

This research work is supported by NSF grant 0406325. We are thankful to the anonymous reviewers for their useful feedback.

## References

1. C. Rigney, S. Willens, A. Rubens, W. Simpson: Remote Authentication Dial In User Service (RADIUS): IETF RFC 2865, June 2000.
2. Eduardo B. Fernandez, Reghu Warriar: Remote Authenticator/Authorizer: In Proceedings of Pattern Languages of Programs (PLoP 2003).
3. David F. Ferraiolo, Janet A. Cugini, D. Richard Kuhn: Role based Access Control (RBAC): Features and Motivations: Proceedings of Computer Security Application Conference, December 1995, pp. 241-248.
4. Narendar Shankar, William A. Arbaugh: On trust for Ubiquitous Computing: In Proceedings of Workshop on Security for Ubiquitous Computing, 2002.
5. Kimmo Janhunen: Recent developments in Authentication, Authorization and Accountancy:

6. Nissanka B. Priyantha, Anit Chakraborty, Hari Balakrishnan: The Cricket location support system: Proceedings of the 6th ACM MOBICOM, Boston, MA, August 2000.
7. Paramvir Bahl, Anand Balachandran, Allen Miu, Wilf Russell, Geoffrey M. Voelker, Yi-Min Wang: PAWNS: Satisfying the Need for Ubiquitous Secure Connectivity and Location Services: IEEE Personal Communications Magazine (PCS), Vol. 9, No. 1, February 2002.
8. Philippe Debaty, Debbie Caswell: Uniform Web Presence Architecture for People, Places and Things: IEEE Personal Communications 8(4) (August 2001) 611.
9. Tim Kindberg, Kan Zhang, Narendar Shankar: Context authentication using constrained channels: In Proceedings of the Fourth IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 2002), pages 1421, June 2002.
10. Eduardo B. Fernandez, Rouyi Pan: A pattern language for security models: In Proceedings of 8th Conference on Pattern Languages of Programs (PLoP 2001), Monticello, IL, USA, September 2001.
11. Andy Harter, Andy Hopper, Pete Steggles, Andy Ward, Paul Webster: The anatomy of a context aware application: Proceedings of 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking, Seattle, Washington, USA, August 1999, pp. 59-68
12. Roy Want, Andy Hopper, Veronica Falcao, Jonathan Gibbons: The active badge location system: ACM Transactions on Information Systems, vol. 10, pp. 91-102.
13. Riccardo Focardi: Using entity location for the analysis of authentication protocols: In Proceedings of sixth Italian Conference on Theoretical Computer Science (ICTCS 98), Prato, 9-11 November 1998.
14. Simon G. M. Koo, Catherine Rosenberg, Hoi-ho Chan, Yat Chung Lee: Location discovery in Enterprise-based Wireless Networks, Implementation and Application: Second IEEE Workshop on Applications and Services in Wireless Networks (ASWN 2002), Paris. July 2002.
15. Jakob E. Bardram, Rasmus E. Kjaer, Michael Pederson: Context-Aware User Authentication Supporting proximity based login in pervasive computing: Proceedings of Ubicomp 2003: Ubiquitous Computing, volume 2864 of Lecture Notes in Computer Science, pages 107123, Seattle, Washington, USA, Oct. 2003.
16. Linux Kernel 2.6.11: Available at <http://www.kernel.org/>
17. N. Sastry, U. Shankar, D. Wagner: Secure verification of Location Claims: ACM Workshop on Wireless Security (WiSe 2003), September 2003.



# Optimal Scheduling for Networks of RFID Readers

Vinay Deolalikar\*, John Recker, Malena Mesarina, and Salil Pradhan

Hewlett-Packard Labs, Palo Alto CA 94304  
vinayd@hpl.hp.com

**Abstract.** Devising switching schemes for networks of colliding and correlated RFID readers is a core challenge in the deployment of RFID networks. We derive optimal scheduling schemes for readers in RFID networks in four cases of practical importance. Most other cases can be reduced to a combination of these basic cases.

## 1 Introduction

RFID (Radio Frequency Identification) technology is expected to play a central role in asset tracking and management in the near future. RFID networks are being deployed to track the flow of assets through various environments. The central question concerning the efficient functioning of such networks may now be posed as follows. Given the topology of the RFID network, including information on potential collisions between readers, and the correlations between the various RFID readers, what is the optimum scheduling of the RFID readers in the network? Indeed, an answer to this question must lie at the heart of every efficient algorithm to schedule readers in a RFID network. This article answers this question in several scenarios that commonly occur in practical deployments of RFID systems.

Our work seeks to combine approaches and insights from two different areas. First, there is the literature from the collaborative signal and information processing (CSIP) community, such as [6], [7]. However, in the scenarios we examine in this paper, communication and processing resources are not primary bottlenecks. However the end goal of an RFID system is the same: to maximize the amount of information extracted from the RFID tags in the environment.

The second area from which we extract techniques started with the application of graph theoretic tools to the problem of assigning radio frequency spectrum to a set of radio frequency transmitters, such as cellular telephone base stations [4], [5]. This work was latter re-examined in the context of frequency assignment in multi-channel (multi-frequency) RFID reader systems in [2]. Thus graphical models found their way into the analysis of RFID reader collision problems. This effort has been extended in another paper by the authors where they consider perturbations of these graphical models [1].

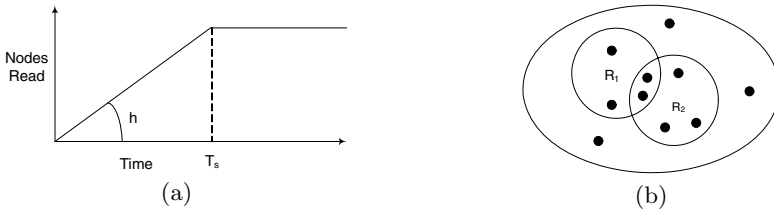
Consequently, the methods of this paper are derived from information processing as well as graph theory.

---

\* Corresponding author.

## 2 RFID Reader Scheduling

**Reader Model:** In this paper, an RFID reader reading a fixed set of tags is modeled as operating in two time regions, an acquisition region, and a saturation region as shown in figure 1 (a). In the acquisition region, a reader has less than the minimum amount of time  $T_s$  necessary for it to read its maximum number of tags. We assume that in the acquisition region, the number of tags read increase linearly with time, till they reach a maximum at time  $T_s$ .  $T_s$  is referred to as the *saturation time* of the reader. In the saturation region, the reader has more than the time  $T_s$  required to read its maximum number of tags.



**Fig. 1.** a) Number of tags read by RFID reader with time. b) RFID reader correlation.  $Corr_{R_1} = \frac{4}{10} = 0.4$ ,  $Corr_{R_2} = 0.5$ , Cross correlation  $Corr_{R_1}^2 = R_1 \cap R_2 = 0.2$ .

*Remark 1.* While we assume a linear approximation to the acquisition region, it can be an arbitrary well behaved function during that time interval and yet be amenable to our method of analysis.

We can now formally state our goal. Let  $T$  be the time period of operation of the RFID network and  $T_i$  be the time allocated to the  $i^{th}$  reader. Then our goal is to maximize the number of distinct tags read under the constraint  $\sum_i T_i = T$ .

To achieve this goal, we start with an elementary, but useful, observation. There is no advantage to giving more time than  $T_{s_i}$  to the  $i^{th}$  reader, since beyond  $T_{s_i}$  it does not read any more tags. It follows that we should operate all readers in their acquisition region. Therefore we impose the constraint  $T_i \leq T_{s_i}$ . If we assume a linear tag read rate in the acquisition region for all the RFID readers in the network, with a slope of  $h_i$  for the  $i^{th}$  reader, the network will read at most  $N$  tags, where  $N \leq \sum_i h_i T_i$ . Equality is achieved when all reads are disjoint.

In the following sections, we will find optimal switching schemes for various simple topologies of RFID readers and tags. We assume that tags are contained inside boxes, which are passing through the RFID reader system. Throughout the rest of this article, we will assume that the following information is either available or the means to compute it are available to us.

1. The interrogation zone overlap for each pair of readers,
2. Each reader’s rate of tag read, and
3. Statistical correlation between each reader’s total tags read at saturation and total tags in a typical box.

### 3 Correlation and Normalization

The use of correlation information is central to our analysis. First we explain what correlation means in the context of RFID readers and how we intend to compute its numerical value.

There are two correlations that we need. The first is the correlation between a reader and the event, which in our case is a box of tags. We model this simply as the fraction of tags in the box that are correctly read by the reader. Cross correlation between two readers is modeled similarly. It is the fraction of tags that are read by *both* the readers. We can extend this to correlation between any number of readers similarly (see figure 1 (b) ).

In practice, what we would do is to pass many boxes through the RFID reader network, and note these fractions for each box and then average out so as to get a statistical correlation that reflects the particular topology and environment of deployment of the network.

One thing that cross-correlation allows us to do is to “normalize” the correlation of a reader to reflect the number of *unique* tags a reader will read from the remaining unread tags in a previously scanned stream of tags. We formalize this notion below.

**Definition 1.** *The normalized correlation of the  $i^{\text{th}}$  RFID reader with respect to a set of readers  $S = \{R_j\}_{j \neq i}$  is a scaling of the original correlation  $Corr_i$  by its cross-correlation with each of the RFID readers in  $S$ . Thus the normalized correlation of the  $i^{\text{th}}$  reader with respect to  $S$  is given by*

$$Corr_i^S = Corr_i \prod_{R_j \in S} (1 - r_{ij}) \tag{1}$$

where  $r_{ij}$  is the cross-correlation between the  $R_i$  and  $R_j$ .

*Remark 2.* Note that in the definition above, the normalized correlation refers to the correlation between a reader and the event.

**Definition 2.** *A virtual reader is a reader whose correlation with an event has been normalized with respect to a fixed set of readers.*

*Remark 3.* The process of creating virtual readers does not change the saturation time  $T_s$  of the reader. Though some reads from virtual readers will prove to be redundant, each reader must still read its full complement of tags.

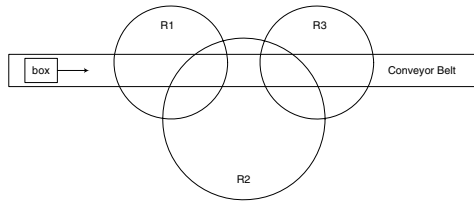
The use of correlation allows us to work around two important problems. First, as noted above, our goal is to maximize the number of *distinct* tags read. Correlation allows us to approach this problem statistically by keeping account of the number of tags that are read in common by sets of readers.

Next, RFID reader system can often be quite complex, with many readers, overlapping interrogation zones, and different path lengths of the tags within each reader’s interrogation zone. Furthermore, the physical characteristics of the readers, box, conveyor belt and tags are highly system dependant. Correlation

allows us to summarize all the system dependant characteristics of the readers in a single number. Furthermore, this number can be determined experimentally for each subset of readers in a system as explained earlier.

### 4 One Box, Multiple Readers

We start by examining the scenario depicted in figure 2. A single box with a number of RFID tags moves along a conveyor belt, through the interrogation zones of several RFID readers. These interrogation zones overlap as shown in the figure. Although the figure shows only three readers, we will see that the algorithm we will provide works for any number of readers.



**Fig. 2.** Single box passing by several RFID readers

Due to the overlaps, operating reader  $R_2$  simultaneously with either  $R_1$  or  $R_3$  will result in a collision and therefore no reads for either of the readers. This rules out the stratagem of keeping all the three readers switched on at all times and simply collating their reads. The objective then is to find the optimal switching scheme for this topology of readers and tags.

Recalling our reader model where the number of tags read is linearly proportional to the reading time up to the saturation time, the total number of tags read is then given by

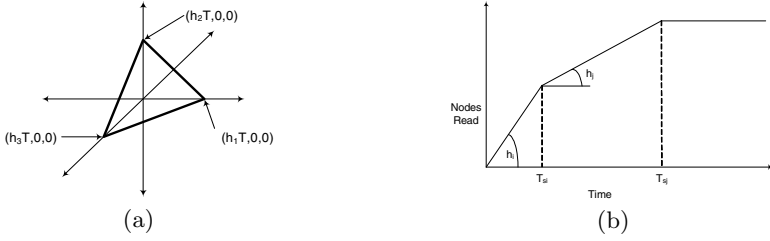
$$N = \sum_i h_i T_i, \quad T_i \leq T_{S_i}. \tag{2}$$

The constraint on time is

$$T = \sum_i T_i \leq \sum T_{S_i}. \tag{3}$$

Equation 2 is the equation of a hyper-plane. And the objective is to find the maximum of this hyperplane function under the constraints given by Equation 3. These constraints define the portion of the hyperplane that lies entirely in the positive quadrant, as shown in Figure 3(a).

The candidates for the maximum value of this function must lie on the boundary, and indeed must be one of the three intercepts with the axes. The largest intercept corresponds to the reader with the highest read rate. Of course, this is true only so far as that reader saturates.



**Fig. 3.** Alternate visualizations of equation 2. a) Hyper-plane for 3 RFID readers. b) Summing contributions from successive readers.

---

**Algorithm 1.** Read Order for the one box, multiple readers problem

---

- 1:  $t \leftarrow$  available time
  - 2: **while**  $t > 0$  **do**
  - 3: Assign available read time up to time  $T_{s_i}$  to reader  $i$  from list of remaining sensors with the greatest slope  $h_i$ .
  - 4: Normalize all correlations with respect to the reader chosen in step 1 so as to yield a set of virtual sensors.
  - 5:  $t \leftarrow t - T_{s_i}$
  - 6: **end while**
- 

The algorithm for assigning read order and time thus proceeds as follows.

Note that there is no need to extend step 1 in the algorithm by firing other readers that do not collide with reader  $R_i$ . This is because there is only one box in the system, and so if it is in the field of  $R_i$ , it cannot possibly be in the field of a reader that does not collide with  $R_i$ , yielding no benefit to firing that reader. This is clearly not the case for multiple boxes.

## 5 Continuous Box Stream, One Reader

Section 4 solved the problem of a single box moving past a collection of RFID readers, some of whose fields overlapped. We will now examine the case of a continuous stream of boxes moving with velocity  $d$  and tag density  $d$  through the interrogation field of a RFID reader  $R_1$ , as shown in figure 4(a). It is obvious that that optimum strategy is to have reader  $R_1$  reading at all times. However, we wish to derive an expression for the number of tags a typical reader can be expected to read.

We first make the observation that for a given tag in a differential region  $dx$  situated  $x$  units into its field, the probability that the reader  $R_1$  has read the tag is

$$P_{dx}(Read) = \left[ \frac{x}{T_{s_1}} \right] Corr_{R_1}. \tag{4}$$

It follows that the average number of tags read in the differential region  $dx$ , denoted by  $TagsRead_{dx}$ , is

$$TagsRead_{dx} = P_{dx}(Read)(dx * d) = \left\lceil \frac{x}{T_{s1}} \right\rceil Corr_{R1}(dx * d). \tag{5}$$

The total number of tags read on the average, denoted  $TagsRead$ , therefore is

$$TagsRead = \int_0^L \frac{Corr_{R1}d}{T_{s1}v} x dx = \frac{Corr_{R1}}{T_{s1}} \frac{d}{2v} L^2. \tag{6}$$

We will use the above expression in the following sections.

### 6 Continuous Box Stream, Two Readers

We now add a second RFID reader to the scenario of section 5 and examine the case of a continuous stream of boxes moving past a pair of readers with completely overlapping fields. This is depicted in figure 4(b). A stream of boxes with tag density  $d$  of boxes on the conveyor belt is passing through the interrogation zones of readers  $R_1, R_2$ . We wish to provide an optimal scheduling for the two readers. Equivalently, we wish ascertain at what points in time the readers should be switched.

Firstly, we use equation 6 to compute the number of tags read by reader  $R_2$ . We obtain

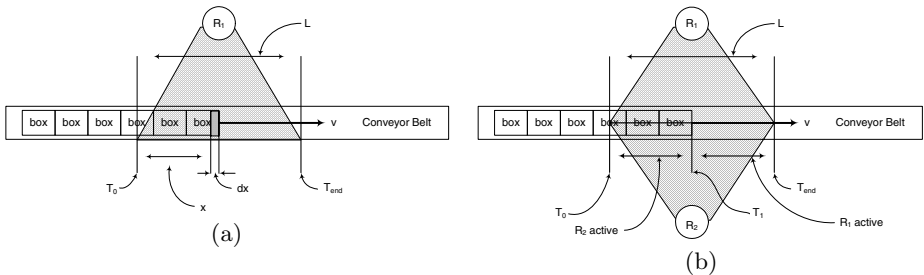
$$R_2read = \frac{Corr_{R2}}{T_{sR2}} \frac{d}{2} (T_1 - T_0)^2. \tag{7}$$

Similarly, the number of tags Reader  $R_1$  would have read had it been active during this initial period is given by

$$R_1read = \frac{Corr_{R1}}{T_{sR1}} \frac{d}{2} (T_1 - T_0)^2. \tag{8}$$

It immediately follows that one should only activate reader  $R_2$  in the period  $[T_0 T_1]$  in one of the following two cases. Either

$$\frac{Corr_{R2}}{T_{sR2}} \geq \frac{Corr_{R1}}{T_{sR1}}, \tag{9}$$



**Fig. 4.** a) The continuous box stream, one reader problem. b) The continuous box stream, two reader problem.

---

**Algorithm 2.** Read Order for the one box, multiple readers problem

---

```

1:  $t \leftarrow$  available time
2: while  $t > 0$  do
3:   Order remaining readers by  $\frac{Corr_{R_i}}{T_{sR_i}}$ .
4:   run first ranked reader  $R_{first}$  from step 3 for time  $\min(T_{sR_{first}}, t)$ 
5:   normalize remaining readers by cross-correlation with reader  $R_{first}$ 
6:    $t \leftarrow t - T_{sR_{first}}$ 
7: end while

```

---

in which case

$$T_1 = \min(T_{sR_2}, T_{end}), \quad (10)$$

or reader  $R_1$  achieves saturation, in which case

$$T_1 = \min(T_0, T_{end} - T_{sR_1}). \quad (11)$$

This result is consistent with Algorithm 1. One first assigns all the available time to the fastest reader up to its saturation time. One then assigns any remaining time to the next fastest *virtual* reader. This algorithm can be applied recursively to scenarios with more than two readers as can be seen in algorithm 2.

## 7 Continuous Box Stream, Multiple Readers

Let us now return to the RFID reader configuration from section 4, but replace the single box with the continuous stream of boxes used in the previous two sections.

At this point, the reader is encouraged to refer to Appendix (section 9) for some background material on the graphical analysis of the collision problem in RFID networks.

The principal difference between this scenario and the single box version arises from RFID reader collision. In the single box regime, turning off a reader did not necessarily result in a loss of information. As a result, it was reasonable to schedule only one reader for reads at any given time. However, such is not the case with this scenario. There are always tags underneath every reader. As a result we wish to have as many readers as possible reading in parallel. Algorithm 3 thus merges the algorithms proposed in sections 4 and 6.

In essence, Algorithm 3 uses graph theory to identify reader sets that can be treated as individual readers. This temporary reader set is then activated as if it were a single reader. Readers that do not complete reading are then returned to the reader pool for the next iteration of the algorithm.

### 7.1 Aggregated Partition Read Rates

Algorithm 3 requires computing the aggregated tag read rate for a set of RFID readers. We describe this computation below.

---

**Algorithm 3.** Solution for the continuous box stream, multiple readers problem

---

- 1:  $t \leftarrow$  available time
  - 2: **while**  $t > 0$  **do**
  - 3: Build a collision graph for the RFID readers in the system, depicting RFID readers as nodes and reader interrogation zone overlaps as edges (see appendix (section 9)).
  - 4: Partition the graph into the fewest possible non-interfering sets.
  - 5: Rank the partitions by their aggregated tag read rate.
  - 6: Activate every reader in the first ranked partition from the previous step, for the time  $T_s$  of the reader with the shortest saturation time.
  - 7: Remove any reader that completed reading in previous step from the collision graph, normalize the remaining readers by their cross-correlation with the removed reader.
  - 8:  $t \leftarrow t - T_s$
  - 9: **end while**
- 

We intend for the aggregated correlation of a set of RFID readers to be, on the average, the fraction of unique tags read by the collection of readers from a collection of tags. With this in mind, we begin by computing the aggregated correlation  $Corr_{aggr}$  for a pair of readers.

$$Corr_{aggr} = Corr_{R_1} + Corr_{R_2} - r_{12} \tag{12}$$

The aggregated read rate is obtained by scaling this with the saturation time of the system of two readers.

$$ReadRate_{aggr} = \frac{Corr_{aggr}}{\min(T_{S_1}, T_{S_2})} \tag{13}$$

The general expression for the aggregated correlation for a set of  $n$  readers  $\{R_i\}_{1 \leq i \leq n}$  is obtained by applying the principal of inclusion and exclusion.

$$Corr_{aggr} = \sum_i corr_{R_i} - \sum_{i < j} r_{ij} + \sum_{i < j < k} r_{ijk} - \dots \pm r_{12\dots n} \tag{14}$$

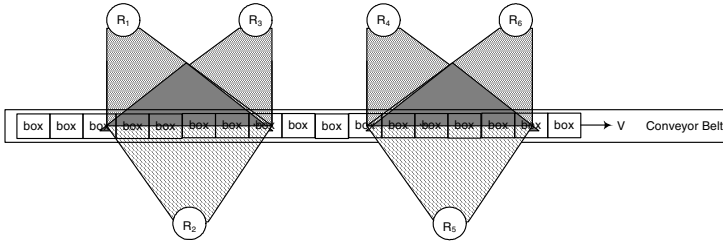
We remind the reader that  $r_{ijk}$  is the fraction of tags read in common by readers  $R_i, R_j$  and  $R_k$  from a set of tags, and so on.

While conceptually simple to evaluate, the aggregate correlation does place additional requirements on the data that must be measured when characterizing a system. In particular, it is not sufficient to obtain the cross-correlation between every pair of readers with overlapping interrogation zones. One must, instead, obtain all 2, 3, ...,  $n$  way cross-correlations in order to perform optimal scheduling of multiple reader RFID systems. This is intuitively satisfying.

## 8 Scheduling Domains

It is not uncommon for readers to be separated by more than their reading range on the same conveyor belt. Such a scenario is depicted in Figure 5. Should each





**Fig. 5.** Two multi-reader configurations on a conveyor belt

of these sets of readers be scheduled separately as independent “scheduling domains”, or are there advantages to scheduling all of these seemingly independent readers as a single block?

We attempt to answer this question using the building blocks developed in the preceding sections to analyze the case of readers which fall into disjoint blocks with no overlaps between blocks.

**Definition 3.** *A scheduling domain is a maximal set of RFID readers such that the collision graph of this topology is connected.*

There are two scheduling domains in Figure 5. The first comprises readers  $R_1$ ,  $R_2$  and  $R_3$ , while the second consists of readers  $R_4$ ,  $R_5$  and  $R_6$ .

Now by the definition of a scheduling domain, readers in distinct scheduling domains do not physically interfere. However, there actually is an advantage to scheduling these readers together. The reason for this is cross domain correlation. While they are physically separate, their aggregated result is affected by the cross-correlation between each of the readers. Take the example of readers  $R_1$  and  $R_4$  in Figure 5. They do not physically interact. However, even if the event correlations of readers  $R_1$  and  $R_4$  with the set of tags are highest in their respective groups, if the cross-correlation of these two readers is very high, there may be an advantage to picking another reader in one of the groups in order to maximize the aggregated number of tags read by the readers.

## References

1. V. Deolalikar, M. Mesarina, J. Recker, D. Das, and S. Pradhan. Perturbative time and frequency allocations for RFID reader networks, preprint 2005.
2. D. Engels, The Reader Collision Problem, Technical Report. MIT-AUTOID-WH-007, 2001. <http://www.autoidcenter.org/research/MIT-AUTOID-WH-007.pdf>
3. F. Harary, Graph Theory, Addison-Wesley 1969.
4. I. Katzela and M. Naghshineh. Channel assignment schemes for cellular mobile telecommunication systems: A comprehensive survey. IEEE Personal Communications, pp. 1031, June 1996.
5. E. Malesinska. Graph-Theoretical Models for Frequency Assignment Problems. PhD thesis, Technischen Universitt Berlin, 1997.

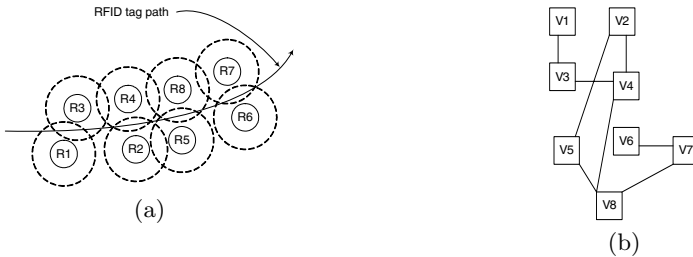
6. K. Yao, R. Hudson, C. Reed, D. Chen, and F. Lorenzelli. Blind Beamforming on a Randomly Distributed Sensor Array System. *IEEE Journal on Selected Areas in Communications*, Vol. 16, No. 8, October 1998.
7. F. Zhao, J. Shin, and J. Reich, "Information-Driven Dynamic Sensor Collaboration", *IEEE Signal Processing Magazine*, March 2002.

## 9 Appendix: The Collision Graph of a RFID Network

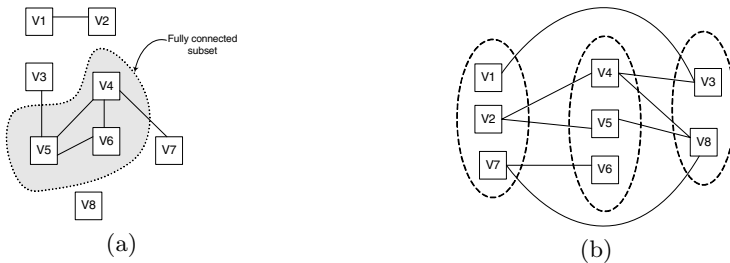
We provide below some elementary facts about the collision graph of a RFID network. For a more detailed graphical analysis of the collision problem for RFID networks, the reader is referred to [1] [2].

An (undirected) graph  $G$  is an ordered pair  $G = (V, E)$  where  $V$  is a set of vertices or nodes and  $E$  is a set of unordered pairs of distinct vertices, called edges. A subset of  $V$  is called an independent set if there are no edges between vertices in  $V$ . An independent set is said to be maximal if the addition of any more vertices will result in a set that is not independent.

A simple undirected graph  $G = (V, E)$  is called bipartite if there exists a partition of the vertex set  $V = V_1 \cup V_2$  where both  $V_1$  and  $V_2$  are independent



**Fig. 6.** RFID Reader partition example. a) RFID reader sample layout. b) Collision graph. Vertices represent the respective RFID reader, edges represent a collision, if both readers are on.



**Fig. 7.** a) The co-graph of figure 6(b), where each edge represents no collision. b) Graph partitioning into independent subsets.

sets. In general, a graph is called  $n$ -partite when its vertex set  $V$  can be written as  $V = V_1 \cup V_2 \dots \cup V_n$ , where  $V_1, V_2, \dots, V_n$  are all independent sets.

Given a collection of RFID readers laid out in some manner, we can construct the associated collision graph  $G = (V, E)$  where each vertex  $v \in V$  corresponds to a RFID reader and each edge  $e \in E$  represents a potential reader field collision between two readers. For example, the collision graph corresponding to the RFID reader layout of Figure 6(a) is given in Figure 6(b). Furthermore, the graph partitioned into independent sets of non-colliding readers as in Figure 7(b).

Readers in any given partition of the collision graph can read simultaneously without interference. Unlike the “one box” problem, there will always be boxes in range every reader, thus it makes sense to fire every reader in a partition when firing one reader in the partition.

# PULSE: A MAC Protocol for RFID Networks\*

Shailesh M. Birari and Sridhar Iyer

K.R. School of Information Technology,  
Indian Institute of Technology, Powai, Mumbai 400 076, India  
{shailesh, sri}@it.iitb.ac.in

**Abstract.** The reader collision problem occurs when the signal from one reader interferes with the signal from other readers. Solutions like RTS-CTS are not applicable because a reader may communicate with multiple tags simultaneously. In this paper, we describe Pulse, a distributed protocol to reduce reader collisions. The operation of the Pulse protocol is based on periodic beaconing on a separate control channel by the reader, while it is reading the tags. The protocol functions effectively with fixed as well as mobile RFID readers. We show, using simulation in QualNet, that using Pulse protocol, the throughput (overall read rate) is increased by as high as 98%(with 49 readers) as compared to “Listen Before Talk” (CSMA) and by 337%(with 9 readers) as compared to Col-orwave. We also present an analytical model for our protocol in a single hop scenario.

## 1 Introduction

An RFID system consists of an RFID reader and a set of RFID tags. The reader uses radio waves to communicate with the tag. A tag may be active (powered by an external battery) or passive (powered by energy in the reader’s signals). Since the signal from a passive tag to the reader is a reflected signal, the read range of a reader is very limited. Not all applications require “always-on”/real-time sensing of the item to be tracked. So a large deployment of fixed readers to cover the area is an overkill. Instead periodic walk-through of fewer mobile readers would suffice to cover the deployment area thus reducing the cost of deployment.

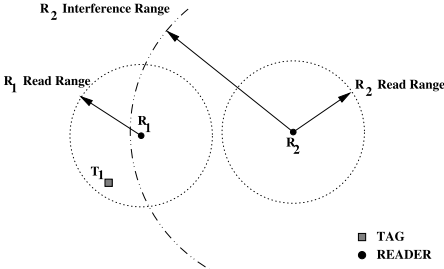
Many applications require readers to operate in close proximity of each other. Due to proximity, the signals from one reader might interfere with the signals from other readers. This interference is called reader collision[1].

**Reader to Reader interference** arises when stronger signal from a reader interfere with the weak reflected signal from a tag. For example, in fig. 1,  $R_1$  lies in interference region of reader  $R_2$ . The reflected signals reaching reader  $R_1$  from tag  $T_1$ , can easily get distorted by signals from  $R_2$ . Note that such interference is possible even when the read range of the two readers do not overlap.

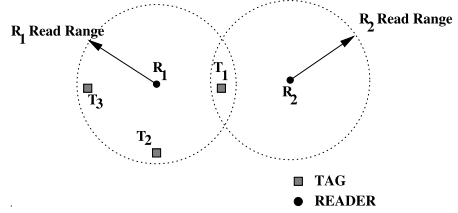
**Multiple reader to tag interference** arises when more than one reader try to read the same tag simultaneously. In fig. 2, the read range of the two readers overlap. Hence the signals from  $R_1$  and  $R_2$  might interfere at tag  $T_1$ . In such

---

\* This work was partly supported by Persistent Systems Pvt. Ltd. [www.persistent.co.in](http://www.persistent.co.in).



**Fig. 1.** Reader to Reader Interference



**Fig. 2.** Reader to Tag Interference

case,  $T_1$  can not decipher any query and the tag is read neither by  $R_1$  nor by  $R_2$ . Due to reader collisions,  $R_1$  will be able to read  $T_2$  and  $T_3$  but it may not be able to read the tag  $T_1$ . In such case,  $R_1$  will indicate presence of 2 tags instead of 3.

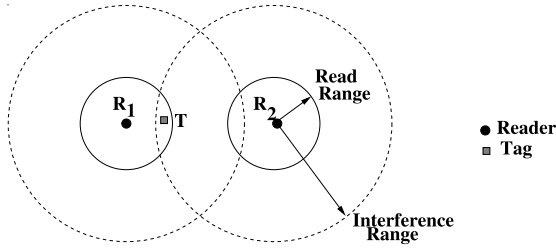
Apart from such incorrect operations, reader collisions also result in reduction of the overall read rate of the RFID system. Hence reducing these reader collisions is essential. Moreover this problem is aggravated in case of mobile/handheld readers.

Standard multiple access mechanisms cannot be directly applied to RFID systems due to the following reasons.

- *FDMA*: With FDMA, the interfering readers use different frequencies to communicate with the tags. Since the RFID tags do not have any frequency selectivity, they cannot select a particular reader frequency for communication. Hence FDMA is not a practical solution in RFID systems.
- *TDMA*: With TDMA, the interfering readers are allotted different time slots thus avoiding simultaneous transmissions. However this is similar to the well known coloring problem in graph theory[1] which is an NP-hard problem[1]. Also because of mobility, non interfering readers may move closer and start interfering making TDMA an inefficient solution.
- *CSMA*: In fig. 3, the read ranges of the two readers donot overlap. However, the signals from reader  $R_2$  can interfere with the signals from reader  $R_1$  at tag T. This case can also happen when the two readers are not in each other's sensing range making carrier sensing(and hence CSMA) ineffective in RFID networks.
- *CDMA*: CDMA will require extra circuitry at the tag which will increase the cost of the tags. Also code assignment to all the tags at the deployment site may be a complicated job.

FDMA, TDMA and CSMA are discussed in more detail in section 2. Standard anti-collision protocols like RTS-CTS cannot be directly applied to RFID systems due to following reasons.

- In case of traditional wireless networks, only one node has to send a CTS back to the sender. However in RFID, if a reader broadcasts an RTS, all



**Fig. 3.** Reader Collision making carrier sensing ineffective

tags in the read range need to send back a CTS to the reader. This demands another anticollision mechanism for these CTS which will make the protocol more complicated.

- Also there are chances that a tag (say  $T_1$ ) may not receive an RTS due to collision while other tag (say  $T_2$ ) may receive it. In such case, a CTS from  $T_2$  is not a guarantee that there is no collision in the read range of the reader.

We propose a distributed protocol, Pulse, based on a beaconing mechanism. While a reader is reading the tags, it periodically broadcasts a beacon on a separate control channel. Any other reader that wants to communicate with the tags, first senses the control channel for a beacon. If it does not receive any beacon for a specified amount of time, it transmits a beacon and starts communicating with the tags. It then continues to periodically transmit a beacon as long as it is communicating with the tags.

## 2 Related Work

The **Class 1 Generation 2 UHF standard**[2] ratified by **EPCGlobal**[3] uses spectral planning(FDMA). It separates the reader transmissions and the tag transmissions spectrally such that tags collide with tags but not with readers and readers collide with readers but not with tags. Such separation solves the reader to reader interference since the reader transmissions and tag transmissions are on separate frequency channels. However the tags donot have frequency selectivity. Hence when two readers using separate frequency communicate with the tag simultaneously, it will lead to collision at the tags. Thus multiple reader to tag interference still exists in this standard.

**Colorwave**[4] is a distributed TDMA based algorithm, where each reader chooses a random time slot to transmit. If it collides, it selects a new timeslot and sends a kick to all its neighbours to indicate selection of new timeslot. If any neighbour has the same color, it chooses a new color and sends a kick (small control packet) and this continues. If the percentage of successful transmission goes below certain threshold, the maxColors is incremented and if the percentage increases beyond certain threshold, the maxColors is decremented. More details about the algorithm can be found in [4].

Colorwave requires time synchronisation between readers. Also, Colorwave assumes that the readers are able to detect collisions in the RFID system. However it may not be practical for a reader alone to detect the collisions that happen at the tags unless the tags take part in the collision detection.

**ETSI EN 302 208**[5][6] is an evolving standard being developed for RFID readers. It has a CSMA based protocol called “Listen Before Talk”. The reader first listens on the data channel for any on-going communication for a specified minimum time. If the channel is idle for that time, it starts reading the tags. If the channel is not idle, it chooses a random backoff. However as described earlier, the readers may not be able to detect collision by carrier sensing alone.

### 3 Pulse Protocol

RFID networks also suffer from the hidden terminal problem. As seen in figure 3,  $R_1$  and  $R_2$  are not in each other’s sensing region, but signals from  $R_2$  might interfere with signals from  $R_1$  at tag  $T$ . For such a scenario, a notification mechanism is required between  $R_1$  and  $R_2$  such that  $R_2$  is informed of  $R_1$ ’s transmissions before it communicates with the tag. We propose to have this notification through a broadcast message called “beacon” on a separate control channel.

The communication range in the control channel is such that, any two readers that can interfere with each other on the data channel (channel used to read the tags), are able to communicate on the control channel. Thus in fig 3, since  $R_1$  and  $R_2$  interfere with each other on the data channel, they will be able to communicate on the control channel. This can be achieved by making the readers transmit at a higher power on the control channel than the data channel. The control channel can simply be a sub-band in the RFID spectrum apart from those used for reader-tag communication. Hence transmission on the control channel will not affect any on-going communication on the data channel.

#### 3.1 Description

Pulse protocol is present only at the reader since the tags do not take part in the collision avoidance. The data channel is used for reader-tag communication whereas the control channel is used for reader-reader communication. We assume that the reader is able to simultaneously receive on both the control and the data channel.

Following is an overview of the Pulse protocol.

- Before communicating with the tags, a reader has to wait in the state *WAITING* for a minimum time  $T_{min}$  which is thrice the beacon interval. The time  $T_{min}$  is analogous to the DIFS time in 802.11 protocol[7]. Everytime it receives a beacon in this state, it resets its waiting time to  $T_{min}$ .
- After  $T_{min}$  time has elapsed and it did not receive any beacon, the reader concludes that there is no other reader in the neighbourhood which is reading the tags. Hence it enters a contention phase and chooses a random backoff

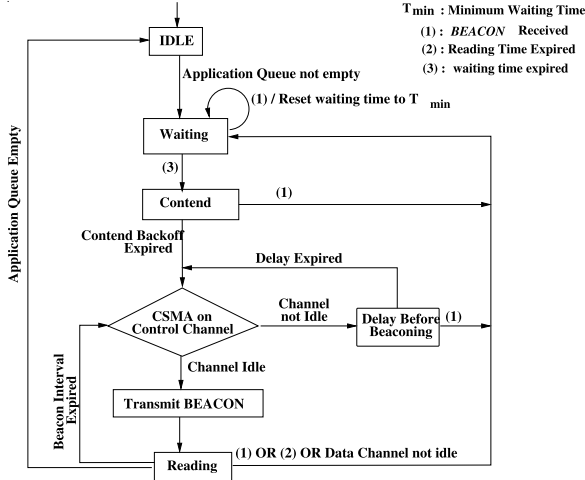


Fig. 4. Flow Chart for Pulse

time (*contend\_backoff*) from the interval  $[0 \dots CW]$ . If it chooses  $i$ , it waits for  $i$  beacon intervals in state *CONTEND*. If it now receives a beacon, it has lost this cycle and waits for the next cycle, i.e until it does not receive a beacon for atleast  $T_{min}$  time. If the randomized backoff time is over and the reader did not receive any beacon, the reader sends a beacon on the control channel and starts communicating with the tags on the data channel. This randomized backoff helps to avoid collisions between readers. *contend\_backoff* is a multiple of beacon intervals to improve fairness.

- While the reader is communicating with the tags, the reader sends a beacon on the control channel every beacon interval. This beacon acts as a notification to the neighbouring readers so that they can withhold their communication with the tags and thus avoid possible collisions. After the communication with the tags is over, the reader again waits in the *WAITING* state and the cycle continues.
- Everytime the reader sends a beacon, it first senses the control channel. If the control channel is busy, it continues to sense the control channel. As soon as the channel gets idle, the reader waits for a random delay (*delay\_before\_beaconsing*) and senses the channel again to send the beacon. This random delay is a multiple of the beacon propagation delay and helps to avoid collisions - otherwise many readers would simultaneously send the beacon after the channel became idle.

Fig. 4 shows the detailed flowchart and fig. 5 shows the detailed algorithm for the Pulse protocol.

The *contend\_backoff* and the *delay\_before\_beaconsing* in the protocol are similar to the backoffs in general wireless networks, they are decreased as long as the control channel is sensed idle, stopped when a transmission is detected, and



```

- CASE: Receive packet from application to send on the network
  1: if state = IDLE then
  2:   state = WAITING
  3:   Set waiting_time_expired timer to  $T_{min}$ 
  4: end if
- CASE: Control channel becomes busy
  1: if state = CONTEND then
  2:   Pause contend_backoff_expired timer
  3: end if
  4: if state = DELAY_BEFORE_BEACONING then
  5:   Pause delay_before_beaconing_expired timer
  6: end if
- CASE: Control channel becomes idle
  1: if state = CONTEND then
  2:   Resume contend_backoff_expired timer
  3: end if
  4: if state = DELAY_BEFORE_BEACONING then
  5:   Resume delay_before_beaconing_expired timer
  6: end if
- CASE: BEACON Received
  1: if state = READING OR state = CONTEND OR state = WAITING then
  2:   Cancel all timers
  3:   state = WAITING
  4:   Set waiting_time_expired timer to  $T_{min}$ 
  5: end if
- CASE: Timer Expired
  1: if waiting_time_expired timer AND state = WAITING then
  2:   state = CONTEND
  3:   Set contend_backoff_expired timer to previous residual value if any else select a new
      random backoff
  4: end if
  5: if (beacon_interval_expired timer AND state = READING) OR
      (contend_backoff_expired timer AND state = CONTEND) then
  6:   if Control channel is IDLE then
  7:     transmit BEACON on control channel
  8:     Set reading_time_expired timer to max allowed communication time, if not set
  9:     Set beacon_interval_expired timer
  10:    state = READING
  11:    Start communication with the tags
  12:   else
  13:    state = DELAY_BEFORE_BEACONING
  14:    Set delay_before_beaconing_expired timer to random delay
  15:   end if
  16: end if
  17: if reading_time_expired timer AND (state = READING OR state =
      DELAY_BEFORE_BEACONING) then
  18:   cancel all timers
  19:   state = WAITING
  20:   Set waiting_time_expired timer to  $T_{min}$ 
  21: end if

```

Fig. 5. Pulse Protocol Algorithm

reactivated when the control channel is sensed idle again. Also, if the reader receives a beacon during backoff (*contend.backoff*), in the contention phase, it stores the residual backoff timer and then waits for the next chance, i.e until it does not receive a beacon for atleast  $T_{min}$  time. It then uses this residual backoff time. This is done only to improve fairness amongst readers. Although the protocol seems to be simple, we show using simulations that it is effective in both static and mobile networks.

## 4 Simulation Experiments

### 4.1 Simulation Model

We have simulated the UHF RFID network in QualNet simulator[8] with data channel frequency as 915MHz and the control channel frequency as 930MHz. We assumed: No inter channel interference between the data and the control channel, Free space propagation path loss, no fading, SNR based signal reception(SNR = 10), omni-directional antennas, 2 Mbps data rate, -91dBm Radio Rx sensitivity and -81dBm Rx threshold, negligible data processing delay and channel switching delay and packet collision as the only cause of packet loss. We also adjusted the transmission power of the RFID node -45dBm, to make the read range  $\sim$  5 feet as is the case with UHF RFID readers.

With these parameters the read range, sensing range and the interference range are 5.31 feet(1.62 meters), 17.71 feet(5.4 meters) and 23.29 feet(7.1 meters) respectively. Here the interference range is the maximum distance upto which a reader's transmission can interfere with another reader-tag communication. Thus the beacon range should be **atleast** equal to the interference range in order to make this protocol effective.

We define the **Beacon Range Factor(BRF)** as the ratio of the control channel transmission power to the data channel transmission power. According to [9], the power received at a receiver is inversely proportional to the square of the distance between the transmitter and receiver. Thus, BRF is given by[10]

$$BRF = \frac{P_{Beacon}}{P_{Data}} = \frac{r_{Beacon}^2}{r_{Data}^2}$$

Thus with data range as 1.62 meters, in order to have a beacon range of 7.1 meters, we require a BRF of 19.2.

### 4.2 Performance Metrics

A query is said to be successfully sent if it is sent by a reader and is successfully received by all the tags in the read range i.e. it does not collide with any other query in the network. We define the system throughput and the percentage efficiency as follows.

$$\text{System Throughput} = \frac{\text{Total queries sent successfully (by all readers)}}{\text{Total time}}$$

$$\text{System Efficiency (\%)} = \frac{\text{Total queries sent successfully (by all readers)} \times 100}{\text{Total queries sent (successful + collided) by all readers}}$$

In general, the tag identification is through a query-response protocol where the reader sends a query and the tag responds with its unique identification number. Higher the number of queries sent successfully, higher the throughput, and hence higher would be the number of tags identified by the readers. Percentage efficiency reflects the ability of a protocol to detect a possibility of collision at the tags and hence avoid unnecessary transmissions. An improvement in throughput

indicates an improvement in the read rate whereas an improvement in the efficiency indicates reduction in collisions. Thus throughput and efficiency together define the effectiveness of the protocol. Through simulations we show that Pulse protocol is effective in both the dimensions.

### 4.3 Simulation Scenarios

We used the following simulation setup for running the experiments.

- *Tag setup*: We used a field of 10 meter X 10 meter area, with 400 tags forming a grid of 20 X 20. The tags were placed throughout the simulation field with 0.5 meter interval so that most of the collisions in the field would be detected by these tags.
- *Fixed Readers*: For fixed reader simulation, all the readers were randomly placed in the field. We used 20 random topologies with 3 different seeds in each case giving a total of 60 simulations per protocol.
- *Mobile Readers*: For simulation of mobile readers, the initial placement of readers was a uniform grid of readers. We used a random way point mobility with low speed of 0.5 to 2 meters per second and 10 random seeds.

For simulation, the RFID application generated a packet(query) to be sent to the tags with exponential interarrival time of average 500  $\mu$ sec throughout the simulation time of 60 seconds.

### 4.4 Compared Protocols

We compared our Pulse protocol with Aloha protocol, CSMA protocol[5][6] and Colorwave. A reader with Aloha protocol assumes that it is the only reader communicating with the tag. Hence when the reader wants to communicate with the tags, it simply starts its transmission without applying any collision avoidance. The CSMA protocol is similar to **ETSI EN 302 208**[5][6] with a listen time of 15msec. For Colorwave protocol, we used the time slot of 10 msec. Rest of the experiment setup for Colorwave was as given in [4]

We set the beacon interval of Pulse protocol as 5 msec and  $T_{min}$  same as the listen time in CSMA i.e 15msec. Using similar settings for both the protocols help us evaluate the MAC protocols in an unbiased manner.

## 5 Results

Keeping BRF=28 and beacon interval=5msec, we did the comparison initially on a 25 reader topology followed by topologies with different number of readers (4..64). We also studied the effect of BRF and beaconing interval on throughput and efficiency of Pulse.

### 5.1 Throughput

*25 Reader Topology*: Fig. 6 shows system throughput with different protocols.

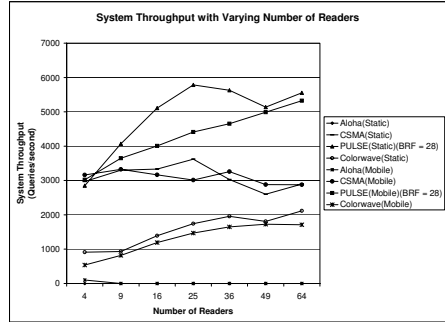
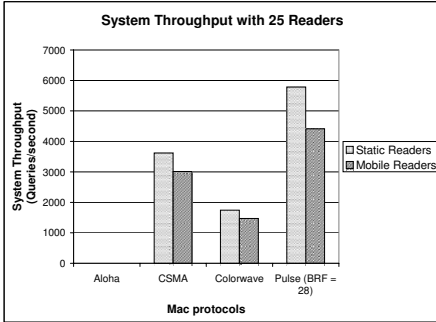


Fig. 6. Throughput comparison with 25 readers

Fig. 7. Throughput comparison with different number of readers

- With Aloha protocol, almost every transmission in the system collided because readers with aloha protocol do not apply any collision avoidance.
- CSMA has better throughput than Aloha because carrier sensing is successful in avoiding collision with readers within the sensing range. However number of collisions using CSMA is still high due to the hidden terminal problem.
- In colorwave, because of the distributed timeslot mechanism, the timeslots are underutilised thus showing lower throughput.
- In Pulse, these collisions are avoided because the beacon sent by a reader acts as a notification to the neighbouring readers(including hidden nodes), which then withhold their transmission thus avoiding collisions. Pulse shows throughput improvement of 60% as compared to CSMA and 232% as compared to Colorwave in static topology.
- Even in case of mobility, Pulse remains to be effective with throughput improvement of about 46% as compared to CSMA and 200% as compared to Colorwave.

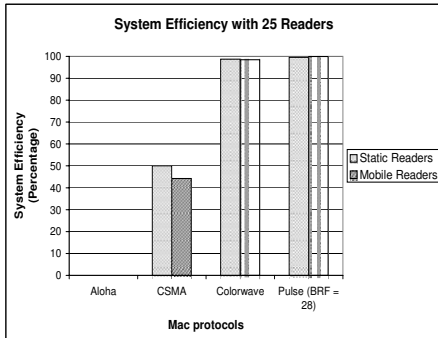
*Varying Number of Readers:* Fig. 7 shows the graph of throughput with varying number of readers in the system. Following are the observations:

- Aloha continues to show negligible throughput.
- As the number of readers in the system are increased, the throughput of CSMA protocol does not increase. Hence unable to cater to dense networks.
- Pulse protocol shows better throughput in all topologies as compared to both colorwave and CSMA protocol. It shows an improvement of as high as 98% (with 49 readers) over CSMA and 337% (with 9 readers) over Colorwave.
- Using Pulse protocol, the throughput of the system keeps on increasing as the number of readers in the system is increased upto a **saturation point** after which the throughput stops increasing even if the number of readers is increased. For example for BRF=28, 25 readers is the saturation point. Hence if the throughput of the system is of prime importance, no more than the saturation number of readers should be deployed.
- Note that Pulse is effective even in a highly dense network of 64 readers.

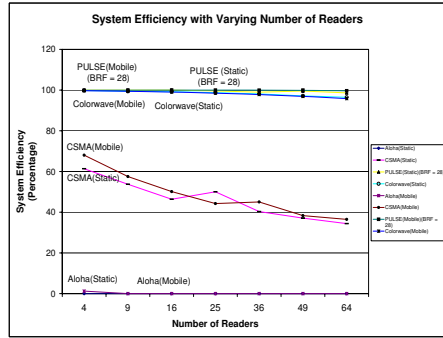
### 5.2 Efficiency

*25 Reader Topology:* Fig. 8 shows system efficiency with different MAC protocols.

- the efficiency with CSMA barely crosses 50% which means that 50% of the transmissions in the network are wasted due to collision.
- Using Colorwave, the efficiency is almost 100% however, colorwave fails to give better throughput than Pulse.
- With Pulse, the efficiency is above 99% with both static and mobile reader network. Thus Pulse is successful in detecting possibility of collisions and thus avoid the same.



**Fig. 8.** Efficiency with 25 Readers



**Fig. 9.** Efficiency with Varying Number of Readers

*Varying Number of Readers:* Fig. 9 shows the graph for the same.

- With Aloha protocol, the efficiency is negligible in all the experiments.
- As seen, the efficiency of CSMA keeps on decreasing as the number of readers go on increasing. As the density of the network increases, the number of hidden terminals increase thus reducing the efficiency.
- Pulse protocol overcomes the hidden terminal problem through a beacon and hence the efficiency of the system is above 95% in all topologies.

Thus Pulse is definitely an improvement over the existing solutions in both the dimensions of throughput and efficiency. It remains to be effective even in a highly dense mobile network.

We further tested the effect of the protocol parameters, BRF and beaconing interval, on the system throughput and efficiency. We found that BRF of 28 gives the highest throughput. We also found that change in beacon interval does not show any significant change in the system throughput. We also theoretically analysed Pulse for which we assumed that all the readers in the network are in each others’ communication range on the control channel. We divided the time into cycles and calculated the average number of queries transmitted per cycle which gives the overall system throughput. More details can be found in [10]. ‘

## 6 Conclusion and Future Work

The reader collision problem in RFID networks is a hindrance for the proliferation of RFID. We presented a distributed protocol, Pulse, for an RFID network which uses a beaconing mechanism by sending periodic beacon on the control channel. Although the protocol is simple, we have shown that it mitigates the reader collision problem. It reduces the reader collisions to 1-2% and also increases the read rate of the system by as high as 98% as compared to CSMA. It requires very less overhead on the reader side and absolutely no support on the tag side. Our protocol is also very effective in a mobile scenario facilitating the use of mobile readers which is a cost effective solution for many applications.

We did not account for any channel switching delay in our simulations. However we believe it to be negligible as compared to the beacon interval. Ofcourse, the Pulse protocol demands for some extra circuitry on the receiver end of a reader. However Pulse protocol increases the throughput considerably. It also promotes the use of lesser number of readers by being effective in a mobile scenario. We believe this performance gain and reduction in number of readers required is high enough to offset the hardware modification required by this protocol.

Further research can involve porting of the Pulse protocol to readers with multiple data channels. Further analysis will lead to insights on the ideal parameters like the beaconing interval, waiting time and the maximum capacity of the protocol.

## References

1. Daniel W. Engels. The reader Collision Problem. Technical report, epcglobal.org, 2002.
2. Chris Diorio. Class 1 generation 2, uhf rfid. CTAN: [www.autoid.org/SC31/2004/dec/SG3\\_200411\\_430\\_Gen2Update.pdf](http://www.autoid.org/SC31/2004/dec/SG3_200411_430_Gen2Update.pdf), December 2004.
3. Electronic Product Code. <http://www.epcglobalinc.org>.
4. J. Waldrop, D. W. Engels, and S. E. Sarma. Colorwave: An Anticollision Algorithm for the Reader Collision Problem. In *IEEE Wireless Communications and Networking Conference (WCNC)*, 2003.
5. *ETSI EN 302 208-1 v1.1.1*, September 2004. CTAN: <http://www.etsi.org>.
6. *ETSI EN 302 208-2 v1.1.1*, September 2004. CTAN: <http://www.etsi.org>.
7. IEEE Computer Society LAN MAN Standards Committee. Wireless LAN Medium Access Control (MAC) and Physical Layer(PHY) Specifications. In *ANSI/IEEE Std. 802.11, 1999 Edition, The Institute of Electrical and Electronics Engineers*. New York, 1999.
8. Qualnet Simulator 3.6. <http://www.qualnet.com>.
9. Prashant Krishnamurthy Kaveh Pahlavan. *Principles of Wireless Networks*. Pearson Education, 2002.
10. Shailesh Birari. Mitigating the Reader Collision Problem in RFID Networks in Mobile Readers. Master's thesis, Indian Institute of Technology, Bombay, July 2005.

# *RFIDcover* - A Coverage Planning Tool for RFID Networks with Mobile Readers

S. Anusha and Sridhar Iyer

Indian Institute of Technology Bombay, India  
{anusha, sri}@it.iitb.ac.in

**Abstract.** Radio Frequency Identification (RFID) finds use in numerous applications involving item identification and tracking. In a typical application, RFID tags are attached to the items and are periodically queried by readers. Using a fixed placement of readers to guarantee complete coverage of all tags in a given area at all times increases the deployment costs. Also, most practical applications do not need complete coverage at all times. It is enough to provide complete coverage periodically, say each tag being covered every  $\tau$  seconds. For such applications, using mobile readers to cover the area would be more cost-effective.

Given an area to be covered completely within a period  $\tau$ , determining the number of mobile readers required, their placement and movement pattern, is a difficult problem. We have developed *RFIDcover*<sup>1</sup>, an automated coverage planning tool, that addresses this problem. Given an application scenario and reader specifications, *RFIDcover* determines an optimal number of readers required to guarantee complete coverage within the specified period  $\tau$ . It also generates a layout giving the placement and movement pattern of the readers. The architecture of *RFIDcover* is generic and extendible, making it easy to implement different application scenarios. In this paper, we present *RFIDcover* implementation for a retail inventory tracking application scenario and evaluate its effectiveness.

## 1 Introduction

Radio Frequency Identification technology finds a plethora of applications in various commercial sectors for tracking and identification of objects. The key components of an RFID system are the tags and readers. The RFID tag is a low functionality microchip with an antenna connected to it, that is attached to the item to be tracked, or identified, and stores the unique identification number of the item. The readers communicate with the tags for reading/writing the information stored on them. Readers may be fixed (stationary at a location) or mobile (within the given area).

The tags used for most applications are *passive*[1] [2], which puts a limit on the readers' *interrogation range* - the range within which a tag can be read by the

---

<sup>1</sup> This work was partly supported by Persistent Systems Pvt. Ltd.  
<http://www.persistent.co.in>

reader. For example, RFID readers operating in the UHF band typically have an interrogation range of 3-5 m[2]. As a result, a large number of readers may be required to provide complete coverage for a given area, leading to significant deployment costs.

Some applications, such as retail inventory tracking, need complete coverage only periodically, say each tag being covered every  $\tau$  seconds. For such applications, using mobile readers to cover the area would be more cost-effective. However, before deploying the readers, it is necessary to answer many important questions, such as: (i) how many readers are needed for providing complete coverage, (ii) where should the readers be placed, (iii) how should the mobile readers move and with what velocity, (iv) how does the number of readers required vary with increase, or decrease, in  $\tau$ . Thus, given an area to be covered completely within a period  $\tau$ , determining an optimal number of mobile readers required, their placement and movement pattern, is a difficult problem.

We have developed *RFIDcover*, an automated coverage planning tool that addresses this problem. Given an application scenario and reader specifications, *RFIDcover* determines an optimal number of readers required to guarantee complete coverage within the specified period  $\tau$ . It does this by: (i) automatically generating a set of layouts (the placement and movement pattern of the readers), (ii) computing the performance metrics - *Cost of deployment* (as a function of the number of readers) and *Tag Reading Time (TRT)* (the time taken to read all tags in the given area) - for each layout, and (iii) selecting the layout which is optimal in terms of both.

The architecture of *RFIDcover* (Section 3) is generic and extendible, making it easy to implement different application scenarios. We consider the following retail inventory tracking application: An RFID tag is attached to each item in a supermarket. The items are then stacked up on the shelves separated by aisles. Periodic inventory checks are carried out using mobile readers moving along the aisles, reading the tags on the shelves as they move. We have implemented *RFIDcover* for such an application (Section 4), and evaluated its effectiveness (Section 5). To the best of our knowledge, there is no literature on coverage tools for RFID systems with mobile readers.

## 2 The Coverage Problem

In this section, we discuss the problem of providing complete coverage using fixed as well as mobile readers. We assume an interference free environment and a circular range for each reader. We derive theoretical results for the number and placement of readers to completely cover a given area. These results form the basis for the mobility models and heuristics used in *RFIDcover*.

### 2.1 Fixed Readers

Given the minimum-area rectangle (with dimensions  $X \times Y$ ) that encloses the area to be covered, the problem of complete coverage using fixed readers is same



as that of covering this rectangle with a number of fixed size circles, each of radius equal to the readers' interrogation range,  $r$ .

With no overlaps among the readers, the maximum coverage that can be achieved is 0.906899682[3]. Hence, complete coverage is possible only if there is overlap amongst the readers. Kershner, in [4], states that the density of an optimal layout, i.e., the ratio of sum total of the area covered by all the circles to the total area to be covered, would be 1.209. [3] discusses a layout that meets this criteria, and hence, is optimal. The number of readers required for complete coverage using the layout is:

$$F_{opt} = \frac{2\sqrt{3}XY}{9r^2} \quad (1)$$

It can be easily seen that using only fixed readers may not be cost-effective as  $X$  and  $Y$  increase, especially if  $r \ll X$ , and/or,  $Y$ . For example, to cover an area of dimension  $10m \times 10m$  with readers of interrogation range,  $r = 2m$ , the number of readers required would be 10. Whereas, to cover an area of dimension  $50m \times 50m$ , this would shoot up to 241. Hence, we explore coverage using mobile readers.

## 2.2 Mobile Readers

The approach for determining the optimal number of mobile readers to cover a given area is similar to that used for fixed readers, except for the following difference: The area covered by a mobile reader would not be of the shape of a circle of radius  $r$ , and would instead be an ellipse-like shape as discussed in [3]. As in the case of fixed readers, complete coverage can be guaranteed only if overlaps are allowed. The number of mobile readers required for one such layout with overlaps discussed in [3] that provides complete coverage is:

$$M = \lceil \frac{X \times Y}{2rv\tau} \rceil \quad (2)$$

Although  $M$  may not be the *minimal* number of mobile readers required, it gives a *sufficient bound* for the number of mobile readers. We note that a deployment using  $M$  readers would need a *to-and-fro* mobility model for each reader. Such a model may be restrictive and impractical for many scenarios. Hence, we use this value of  $M$  only as a comparison point for the evaluation of mobility models implemented in *RFIDcover*. In the next section, we describe the architecture of *RFIDcover* in detail.

## 3 RFIDcover Architecture

The architecture of *RFIDcover* is as shown in Figure 1. It has a three phase operation as follows:

1. *Selection Phase* - In this phase, the mobility model for the mobile readers and the MAC mechanism to be used by the readers for shared access to the medium, are chosen based on the application scenario. Depending on these two, an appropriate heuristic for layout generation is selected.
2. *Generation Phase* - In this phase, the selected heuristic is used to generate a set of possible layouts, each of which conforms to the input constraints and also completely covers the given area. The performance metrics, viz., the *Cost* (as a function of the number of readers), and the *TRT* (total time taken to read all the tags in the entire area) are computed for each such layout.
3. *Optimization Phase*. In this phase, an appropriate objective function for optimization is chosen and applied to the set of layouts generated earlier. This results in the selection of an optimal layout, which is recommended to the user.

An additional feature of *RFIDcover* is that the user can also provide constraints on the *Cost* and *TRT* as an input to the generation and optimization phases.

The following subsections briefly describe the different components shown in the architecture and their roles, using the supermarket inventory application mentioned earlier (Section 1).

### 3.1 Inputs

The user provides as input to *RFIDcover* the following:

- *Reader Specification* - This gives the details (as shown in Figure 1) of both fixed and mobile readers.
- *Topology Specification* - The dimensions of the minimum-area rectangle enclosing the region to be covered and the tag density distribution is provided.
- *Application Scenario* - The application scenario is chosen from a list of those currently supported by *RFIDcover*.
- *Input Constraints* - Additional constraints may be provided by the user which can be modified on-the-fly to generate a new set of layouts.

### 3.2 Functional Overview

Brief description of the functions of each of the basic components in *RFIDcover* is given below. The various components have been loosely grouped together to correspond to the three phases of its operation.

**Selection Phase.** The two main components in this phase are:

- *Model Selector* : The model selector maps the given application scenario and tag distribution to a suitable mobility model and an appropriate MAC mechanism (medium access control). It then passes on the selected mobility model and MAC mechanism as an output to the next phase. The model selector uses the following components for the mapping:

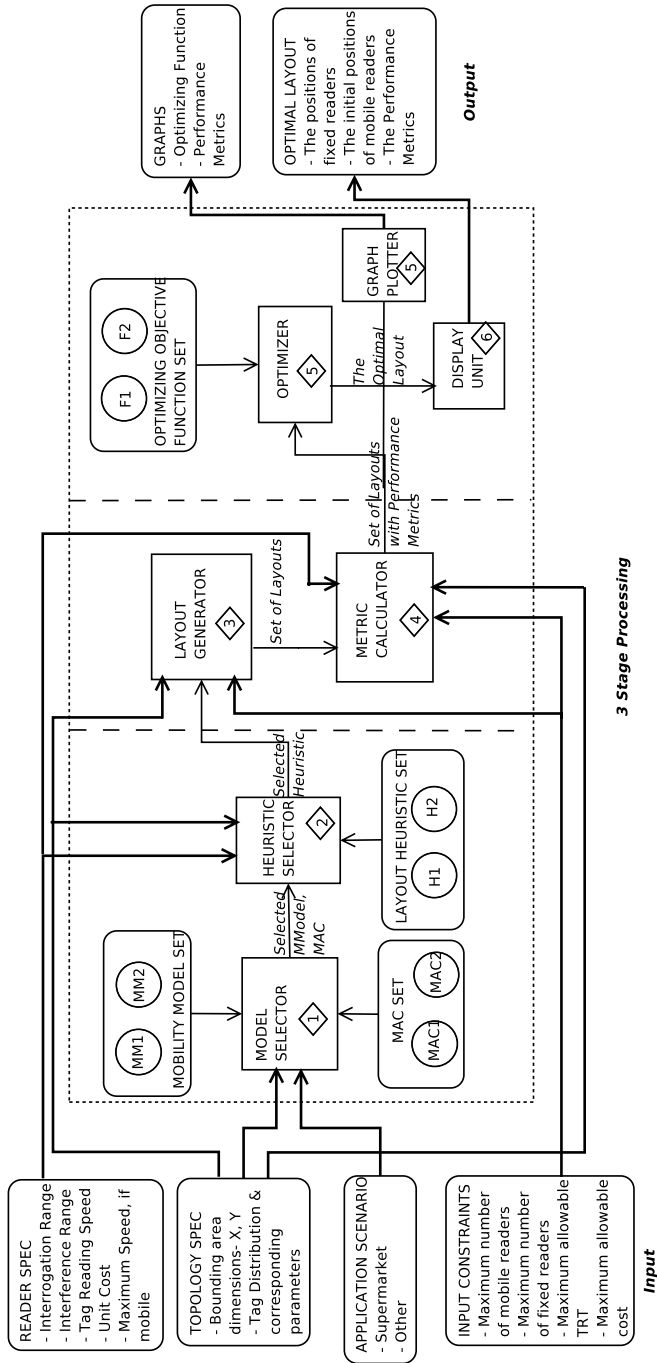


Fig. 1. RFIDcover Architecture

- *Mobility Model Set* : This is a collection of mobility models. A mobility model defines how the mobile readers would move and with what velocity. We assume a homogeneous system, where all the mobile readers follow the same mobility model. A given mobility model may be suitable for some application scenarios and may not be useful for others. Also, note that a mobile reader can not move faster than  $TRS/(2r \times Tag\ density)$  in order to cover all tags in the range.
  - *MAC Set* : The readers may use a number of Medium Access Control (MAC) mechanisms, such as TDMA, FDMA, CSMA, in order to share the medium and read the tags efficiently. MAC set is the collection of such mechanisms currently supported by *RFIDcover*. Any new MAC mechanism can be implemented and added to this set.
- *Layout Generating Heuristic Selector* : Once the mobility model and MAC mechanism has been fixed, and given the tag distribution, reader specifications and the dimensions of the area to be covered, the layout generating heuristic selector finds an appropriate heuristic to be used for generating the layouts. These layouts provide complete coverage of the area and conform to the input constraints. The layout generating heuristic selector uses the following component for its section decision:
- *Layout Generating Heuristic Set* : This is a collection of heuristics that can be used for generating the layouts. Each heuristic has associated with it some mobility models that it supports. Hence, a heuristic along with the mobility model would result in a layout for complete coverage of the area. For example, a layout generating heuristic could place fixed readers at four corners of the area to be covered and place one mobile reader, initially at left top corner, and let it move in a pattern that ensures coverage of the area.

**Generation Phase.** At the end of the *selection phase* we have all the information needed for the *generation phase*, for generating the layouts for complete coverage. This phase consists of two steps and uses the following components:

- *Layout Generator*: It takes as input the topology specifications, the reader specifications, the input constraints, and the chosen layout generating heuristic, and applies that heuristic to generate a set of possible layouts that would provide complete coverage in conformance with the input constraints.
- *Metric Calculator*: For each layout generated by the layout generator, the performance metrics are computed, and keeping in mind the input constraints, a subset of the layouts that conform to the input constraints is generated. The performance metrics include simple generic ones like the number of fixed and mobile readers, the total *Cost* incurred and the *TRT* (or Tag Reading Time, which is the total time it takes to read all tags in the area). In addition, other metrics specific to MAC mechanism or application scenario could also be computed.

**Optimization Phase.** Once the layouts conforming to the input constraints have been generated, the *optimization phase* then determines best layout from the set of those generated. The basic building blocks of this phase are:

- *Optimizer:* Since the layout is generated using a heuristic, there can be a number of layouts that would conform to the input constraints. Various optimizing objective functions are possible. The optimizer applies a suitable optimizing objective function to the layouts generated and recommends the result to the user, along with a summary of the other conforming layouts. The optimizer uses the following component while choosing the objective function.
  - *Optimizing Objective Function Set:* This is a collection of optimizing objective functions. For each layout generated that conforms to the input constraints, we have a set of performance metrics. An optimizing objective function can be applied on some/all of these metrics for determining the most suitable layout. For our supermarket example, we might be interested in recommending a layout that uses the minimum number of readers. Then the objective function should be the “minimum” function and it should be applied on the number of readers.
- *Graph Plotter:* This is used to plot graphs using the layouts conforming to the constraints. The plots reflect the variation of number of fixed readers, mobile readers, *Cost*, *TRT* and other metrics with different layouts. It also displays the optimizing objective function.
- *Display Unit:* This is used to graphically display in detail the optimal layout and the performance metrics associated with it.

### 3.3 Outputs

Following are the outputs that *RFIDcover* provides to the user.

- *Graphs:* The various graphs generated by the graph plotter are shown to the user for analysis.
- *Optimal Layout:* The details of the layout recommended to the user, are shown to the user on a graphical interface.

In the next section, we present the implementation of *RFIDcover* for the retail inventory tracking application.

## 4 *RFIDcover* Implementation

The architecture presented in Section 3 has been implemented in Java. *RFIDcover* currently supports the supermarket inventory application scenario. It implements the zig-zag mobility model and the static coloring MAC that are suitable for the supermarket application. *LGH<sub>1</sub>*, a heuristic specific to the zig-zag mobility model is used to generate the set of layouts. The *least square sum* optimizing function is then applied on the parameters *TRT* and *Cost*, to get the optimal layout that is suggested to the user.

### 4.1 Zig-Zag Mobility Model

In the zig-zag mobility model, the mobile readers move within a rectangular area in a zig-zag fashion. This rectangular area (as shown in Figure 2) forms the basic unit which is replicated throughout the area to be covered. So, the dimensions of this rectangle could be as large as  $X \times Y$ , or as small as *length of aisle*  $\times$  *inter aisle distance*.

This mobility model is suitable for supermarket scenario where tagged items are stacked up on shelves in rows with aisles separating them. The mobile readers move from left to right (or right to left) along the aisle and then move an inter-aisle distance perpendicular to it, and then move from right to left (or left to right) along the aisle and repeat the whole process again. The model is as shown in Figure 2.

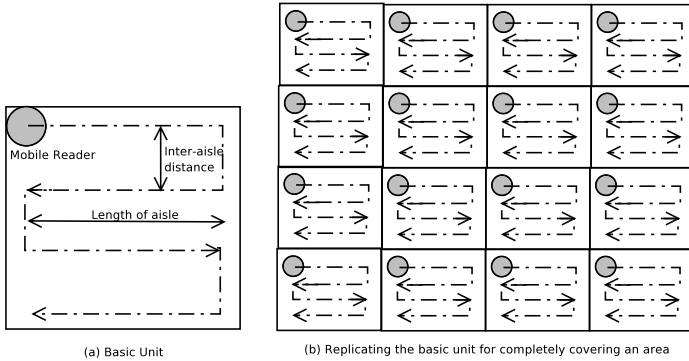


Fig. 2. The Zig-zag Mobility Model

### 4.2 Static Coloring MAC

Static coloring MAC is a TDMA based MAC mechanism for shared access of the readers to the medium of communication. We chose a TDMA based mechanism because this is simple and is commonly used by commercially available RFID readers.

In TDMA based mechanism, each reader is assigned a time slot wherein it can read tags in its range, while the rest of the readers remain silent during the period. Any two readers in the interference range of each other will be assigned different time slots, thus, avoiding any interference or collision. Assigning time slots to readers is analogous to the graph coloring problem [5], where the readers form the vertices of the graph and every pair of interfering readers are adjacent, that is, they have an edge between them. In case of fixed readers, determining these pairs of interfering readers and assigning TDMA slots is straightforward.

In case of mobile readers, determining the number of TDMA slots is not easy, since the readers may move in and out of the interference range of each

other. One simple way to overcome this difficulty is to determine the number of interfering readers in worst case scenario and assign as many slots. The static coloring MAC considers the mobility model, determines all pairs of readers that might come into the interference range of each other, and assigns time slots on a worst case basis. This approach works for many simple mobility models and deterministic layouts, although in general it may be a bit inefficient.

### 4.3 *LGH*<sub>1</sub> Layout Generating Heuristic

*RFIDcover* currently implements a heuristic, which we call *LGH*<sub>1</sub>, for generating the layouts appropriate for the zig-zag mobility model. It generates a hybrid layout consisting of fixed as well as mobile readers. The layouts differ in the strategic points where the fixed readers are placed. *LGH*<sub>1</sub> works as follows: It places fixed readers at the end of every aisle, or at the end of every two aisles and so on, in each direction. The fixed readers, thus, form a grid-like structure over the area to be covered. Each cross section of the grid forms the rectangular region within which one, or more, mobile readers move as per the zig-zag mobility model. More details are given in [3].

### 4.4 Optimization

The layout generating heuristic produces a set of layouts. For each layout, the number of fixed and mobile readers, the performance metrics like the *Cost* and *TRT*, are calculated and those conforming to the input constraints are retained. One of these is chosen and recommended to the user, as per an appropriate optimizing objective function.

In our supermarket application, we would like to use minimum number of readers and yet provide complete coverage as often as possible. Hence, we use the *least square sum* as the objective function and apply it on the parameters *TRT* and *Cost*. Thus, the layout with minimum value for  $TRT^2 + Cost^2$  is chosen as the optimal one.

In the next section, we provide an evaluation of the implementation for the supermarket scenario.

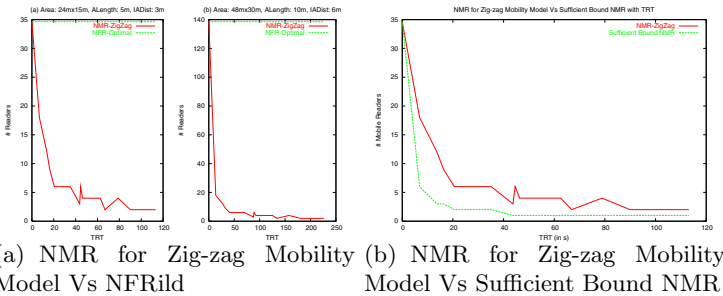
## 5 *RFIDcover* Evaluation for Supermarket Scenario

In this section, we discuss the correctness and usability of *RFIDcover*'s current implementation for the supermarket inventory application scenario. We consider the example shown in Table 1. We ran *RFIDcover* on the example inputs, to get the data used to plot the graphs in Figure 3. Figure 3(a) shows the graphs of the number of mobile readers (NMR) and the optimal number of fixed readers (NFR) with varying *TRT*. From the graphs in the figure, we can see that for providing complete coverage of an area, the number of mobile readers needed is much lesser than the number of fixed readers. The number of mobile readers required decreases drastically as *TRT* increases and would reach 1 when *TRT* approaches  $\infty$ . This is as expected.

**Table 1.** The example

Reader Spec	Topology Spec	Application
Interrogation Range: $2m$ Interrogation Range: $2.5m$ Tag Reading Speed: $70tags/s$ Unit Cost: 1 Max. Speed: $5m/s$	Dimension $X$ : $24m$ Dimension $Y$ : $15m$ Tag Distribution: Uniform Tag Density: $5/m^2$ Aisle length along $X$	Supermarket Scenario Aisle Length: $5m$ Inter Aisle Distance: $3m$
		MAC mechanism
		Static Coloring

For practical applications, very high values of  $TRT$  may not be meaningful. Hence, the region of interest is the area of the graph near the origin, where the  $TRT$  values are within the range of a few seconds. Here also, we find that except when the  $TRT$  is negligibly small, the number of mobile readers required drops significantly with slight increase in  $TRT$ . Hence, using mobile readers for such values of  $TRT$  would be cost-effective.



**Fig. 3.** Observations

Another important observation is that in the region of interest the slope of the number of mobile readers curve falls very steeply if  $r \ll X$ , or  $Y$  but at a much lower rate if  $r$  is comparable to  $X$  and  $Y$ . For example, the curve in the left graph of Figure 3(a) (which corresponds to an area of dimensions  $24m \times 15m$  and  $aisle\ length = 5m$  and  $inter\ aisle\ distance = 3m$ ), falls from 35 readers to around 11 readers as the  $TRT$  increases from  $0s$  to  $15s$ , whereas in the right graph of Figure 3(a) (which corresponds to an area of dimensions  $48m \times 30m$  and  $aisle\ length = 10m$  and  $inter\ aisle\ distance = 6m$ ), it falls from 139 readers to 17 readers, even though the number of aisles is the same in both cases.

The graph in Figure 3(b) compares how the number of mobile readers (NMR) needed for complete coverage using the zig-zag mobility model varies with respect to the *sufficient bound* on the number of mobile readers. As can be observed, the zig-zag mobility model curve follows the *sufficient bound* quite closely. This asserts the practical suitability of the zig-zag mobility model for the supermarket application scenario.



## 6 Conclusions

We have presented *RFIDcover*, an automated coverage planning tool that determines an optimal layout of readers required to guarantee complete coverage for a given application scenario. We have also evaluated the effectiveness of *RFIDcover* for the retail inventory tracking application. *RFIDcover* provides the user with crucial deployment-specific information such as the number of readers needed, their placement, movement pattern, and the deployment cost. It also gives the user the flexibility to input additional constraints on-the-fly, thereby making it a very useful tool for RFID deployment.

The architecture of *RFIDcover* is generic and extendible, enabling easy implementation of other application scenarios. Even for existing application scenarios, other mobility models, MAC mechanisms and layout generating heuristics can be implemented. This would enable a comparison of various deployment options for the same application.

## References

1. Radio frequency identification - a basic primer. White Paper, AIM Inc WP-98/002R2 (2001) <http://www.aimglobal.org>
2. Finkenzeller, K.: RFID Handbook : fundamentals and applications in contactless smart cards and identification. dritte edn. Chichester : John Wiley, Leipzig (2003)
3. Anusha, S., Iyer, S.: *RFIDcover* - a coverage planning tool for rfid networks with mobile readers. Master's thesis, Indian Institute of Technology Bombay (2005)
4. Kershner, R.: The number of circles covering a set American Journal of Mathematics, Vol. 61, 665-671 (1939)
5. Engels, D.W.: The reader collision problem. Technical report, epcglobal.org (2002)

# Vibration Powered Battery-Assisted Passive RFID Tag

Elaine Lai, Andrew Redfern, and Paul Wright

Department of Mechanical Engineering, University of California Berkeley,  
2111 Etcheverry Hall, Berkeley, CA 94720  
{emlai, aredfern, pwright}@kingkong.me.berkeley.edu  
<http://bmi.berkeley.edu>

**Abstract.** Real-time supply chain management, theft prevention, and environmental monitoring motivate the need for RFID systems. Battery-assisted RFID tags increase read range and reliability; however, batteries offer only a finite lifetime solution. Hence, an experiment in powering a battery-assisted passive RFID tag using ambient vibration energy was conducted. A piezoelectric power generator was designed at a resonant frequency of 52 Hertz, with potential power output of 500  $\mu$ W. Manipulation of the electric signal from the generator through a power circuit produced 8mW pulses to the tag, resulting in successful communication with the reader. Power needs were compared to an earlier experiment in powering a Mica2Dot “Mote” showing a 88% decrease in necessary power supply and 90% decrease in charge time.

## 1 Introduction

The benefits of RFID technology are plentiful. Real-time inventory management results in instantaneous communication across the supply chain from manufacturing to retail thus improving on-shelf availability and reducing excess inventory. Automatic and instantaneous visibility of inventory movement helps manage theft prevention. And finally environmental monitoring of perishable goods in transit are essential for customer safety and satisfaction [1].

Passive RFID tags are a reasonable solution in close range applications. However in situations such as environmental monitoring of pallets passing in transit, close range is not always possible. Read range and reliability are improved with battery-assisted passive tags or active tags. However batteries only provide a temporary solution due to their finite life. An alternative solution is to power these tags using ambient vibration energy. Ambient vibrations provide an infinite life and maintenance free solution in powering battery-assisted passive RFID tags.

The following paper details an experiment in using vibrations to power a battery-assisted passive RFID tag. Results of the experiment are compared to earlier experiments in powering a Crossbow Mica2Dot Mote.

## 2 A Battery-Assisted Passive RFID System

The system chosen for implementation is the Alien 2.45 GHz battery-assisted passive tag RFID System. The reasons for choosing this system are: the battery assisted tags

enable longer range, the tags include onboard temperature sensor capabilities which provides a solution for environmental monitoring application, and the tags make use of RF energy from the reader.

The tag battery powers the onboard sensor and electronics including microprocessor and assisting the radio. When a tag is interrogated by the reader, the tag will harness RF energy from the interrogation and use some of that energy in responding to the reader. The RF energy from the reader is harnessed and assists in transmission while the battery helps to increase read range. Information is relayed from tag back to reader through backscatter technology. The signal used to interrogate the tag is modulated and used as a carrier wave for data transmission back to the reader.



**Fig. 1.** Alien's 2.45 GHz Battery Assisted Passive Tag RFID Solution

The results of the Alien system (Alien Technology ALB-2484) have been compared with the results from an earlier experiment for the Crossbow Motes (Crossbow MPR500CA) [2]. The basic functionality of both technologies are the same; they are wireless sensor transceivers. The major differences are that the mote has more memory and greater processing capabilities and does not rely on interrogation from a “reader” to transmit information. Major characteristics are outlined below.

**Physical.** The size of the two wireless sensors are comparable. Both nodes operate with a 3V Coin Cell although a 3V input is not necessarily required.

**Processor.** The RFID tag holds much less memory, 4Kbytes compared to 128Kbytes on the mote. The RFID microprocessor is much more simple and has limited programmability, however current draw is only 1.4mA compared to 8mA for the mote during active mode. During sleep mode, current draw is minimal for both nodes.

**Radio.** The RFID tag is less effective in range at only 30m, one-fifth the range for motes. The benefit of the RFID is that transmission of information takes the form of modulated backscatter. This means that the tag modulates the RF signal from the reader, and when the signal is bounced back, the reader interprets that modulation. Therefore current draw for transmission is greatly decreased, on the tag side. The mote consumes 27mA during transmit and 10mA during receive.

The choice between mote or RFID based system depends on the application. If the user needs the nodes to perform complex tasks, change functionality of the nodes while they are deployed or long range communication, then motes are probably more applicable for the needs of the system. However, if the task at hand is to monitor temperature, pressure, or any other environmental factor, with short range communication then RFID is a sufficient solution. The mote is specified to require 105mW to

sense and transmit, while the RFID is specified to require 4.2mW. Even with the major differences of these two systems they can perform the same basic functionality. An example of a simple system in which either of the two systems may be used is a one hop network, where each of the nodes/tags needs to periodically sample and store the data and then, when interrogated, the data will be transmitted to the base station for processing. The low power usage of RFID, coupled with an energy scavenging power source, is expected to offer a much more reliable long life solution, to low duty-cycle environmental monitoring systems. Hence these experiments shown in Table 1, that compare RFIS with mote performance.

**Table 1.** Comparison of Crossbow Mica2Dot Mote and Alien Technology Battery-Assisted Passive RFID Tag

	Features	Xbow Mica2Dot Mote (MPR500CA)	Alien RFID Tag (ALB-2484)
Physical	Size	2.5Ø x 0.64 cm	8x2.5x0.64 cm
	Cost in bulk	\$25	\$28
	Battery	3V Coin Cell	3V Coin Cell
	Parts	CPU, A/D converter, memory, radio, antenna, sensor, battery	memory, radio, onboard antenna, digital thermometer, battery
Processor	Memory	128K bytes	4K bytes
	Programming	TinyOS	limited
	Current Draw (active)	8 mA	1.4 mA
	Current Draw (sleep)	15 uA	1 uA
Radio	Range	150 m	30 m
	Current Draw (transmit)	27 mA	--
	Current Draw (receive)	10 mA	--
Power	Sense and Transmit	105 mW	4.2 mW
	Sleep	45 uW	3 uW

### 3 “Teeny Temp” System Design

The system design (coined “Teeny Temp”) comprises the following major components: a piezoelectric bimorph generator which converts vibrations to electricity, producing an AC signal, power conditioning circuitry which converts the AC signal from the generator to usable DC signal, and RFID Battery-Assisted Passive Tag with on-board temperature sensor.

The goal of the design is to provide enough power to the RFID system to operate the tag: including powering the onboard sensor, powering the microprocessor, and transmitting useful information to the reader. In order to realize this goal the piezoelectric

power generator must first generate enough power, then the power conditioning circuit must be designed to output a usable supply voltage to the RFID tag while most efficiently making use of the power being generated by the piezoelectric generator.



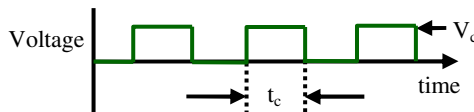
**Fig. 2.** “Teeny Temp” System Design

### 3.1 Power Requirements of the Load

The power requirements of the Alien RFID tag must first be characterized in order to design an optimized power circuit. Three important values need to be determined: critical input voltage, critical pulse length, and current draw. Since we are harvesting minimal amounts of energy, the lowest input voltage and current draw necessary is of value. The critical pulse length must be long enough to reliably wake up the tag, sense temperature, process information, wait for an interrogation from the reader, and transmit tag information.



**Fig. 3.** Alien's Battery-Assisted Passive Tag



**Fig. 4.** Schematic of Input Pulses to Load from Vibration Power Generator

Experiments were conducted using a function generator to input an offset square wave into the RFID tag. Voltage input and pulse length were varied to find critical values. Results are tabulated below. At a read-range of half a meter, the critical operating voltage is 2V and critical length of pulse is 2.5 seconds. Current draw of the tag varies with a maximum of 0.25 mA. As this current value is much lower than the specified value, further design uses the specified value to be cautious.

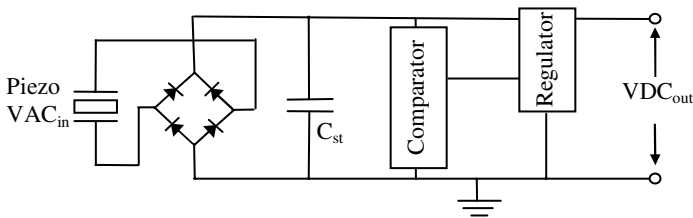
**Table 2.** Specified Versus Experimental Critical Input Values to RFID Tag

		Specifications	Experimental Values
$V_C$	Critical operating voltage	3V	2V
$t_C$	Critical length of pulse	--	2.5 sec
I	Current Draw	1.4 mA	0.25 mA

The critical pulse length of 2.5 seconds is a fairly large number compared to the 0.8 seconds needed by the Crossbow Mica2 Dot Mote. The reason for the longer pulse length with the RFID tag is that the tag waits for an interrogation by the reader before it transmits information while the mote automatically transmits to the base station. Furthermore, the operation of the RFID tag relies partially on harnessed RF energy from the reader. The energy is stored by capacitors onboard the tag and the length of time necessary to charge these capacitors contribute to the critical pulse length.

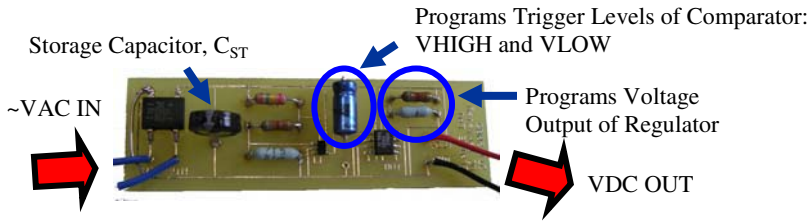
### 3.2 Power Conditioning Circuit

Once the power characteristics of the RFID tag are known, the power conditioning circuit can be built to optimally output the critical values. A schematic of the power conditioning circuit is shown below. The AC signal produced by the piezoelectric power generator is first rectified, then stored in a capacitor. A comparator monitors the voltage level of the capacitor and when it has reached a value,  $V_{HIGH}$ , current is allowed to flow through the regulator. The switch is turned off again when the voltage level of the capacitor has reached a value,  $V_{LOW}$ , to allow the capacitor to charge again.



**Fig. 5.** Schematic of Power Conditioning Circuit with Regulator

The functional requirements of the RFID tag include an input voltage of 2V for a critical pulse length of 2.5 seconds at a current draw of 1.4 mA. The design parameters to be modified on the power conditioning circuit in order to achieve those requirements are storage capacitance,  $C_{ST}$ , trigger levels of the comparator,  $V_{HIGH}$  and  $V_{LOW}$ , and the output voltage of the regulator,  $V_{OUT}$ . The latter three values are programmable via resistors as shown in the figure below. Equipment used include the Maxim MAX6433 comparator and a Texas Instruments TPS72501 voltage regulator.



**Fig. 6.** Power Conditioning Circuit with Regulator

Of the four design parameters,  $V_{OUT}$  is simple to determine and is equal to the critical input voltage of the RFID tag.

$$V_{OUT} = 2V. \quad (1)$$

The next design parameter simple to determine is the low trigger level of the comparator,  $V_{LOW}$ . This trigger level must meet the minimum required input voltage level specified for the regulator in order to achieve a voltage output of 2V. Otherwise, a voltage level not high enough to output 2V on the other side of the regulator becomes wasted energy.  $V_{LOW}$  can be determined from the specification sheets of the regulator.

$$V_{LOW} = 2.5V. \quad (2)$$

The maximum current draw of the RFID tag is specified to be 1.4 mA. Therefore at a critical input voltage of 2V, critical operating power of the RFID tag is 2.8 mW. At a critical pulse length of 2.5 seconds, the energy demanded by the tag is 7 mJ.

$$\text{Power} = VI = 2.8\text{mW} . \quad (3)$$

$$\text{EnergyDemand} = Pt = 7\text{mJ} . \quad (4)$$

The power conditioning circuit, through the combined efforts of the storage capacitor and trigger levels of the comparator, therefore must supply at least 7 mJ of energy to the RFID tag within a pulse of 2.5 seconds. The energy transferred from the storage capacitor can be found using the following equation.

$$\text{EnergySupply} = \frac{1}{2} C \cdot V_{HIGH}^2 - \frac{1}{2} C \cdot V_{LOW}^2 \geq 7\text{mJ}. \quad (5)$$

In order to achieve such a long discharge time, super capacitors were thought to be the best option. The smallest capacitance, well-suited capacitor found was a 22,000  $\mu\text{F}$  gold capacitor, a Panasonic Electric Double Layer Gold Capacitor (SD Series).

$$C_{ST} = 22,000\mu\text{F}. \quad (6)$$

With a capacitance chosen,  $V_{HIGH}$  can be determined.

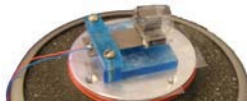
$$V_{HIGH} = 2.63V. \quad (7)$$

The upper trigger level of the comparator must be at least 2.63V if the super capacitor is able to discharge over 2.5 seconds.

The design parameters for the power conditioning circuit have therefore been determined and will be modified upon experimentation as needed.

### 3.3 Piezoelectric Power Generator

The last component of the system to be designed is the piezoelectric power generator. In a real-life application-specific deployment, the power generator would be designed to resonate at the peak frequency of the chosen vibration source. However, experimentation for powering the RFID tag will only be conducted in a lab setting, therefore resonant frequency of the generator was not significant. Although experiments were conducted in lab, a setup was created to mimic characteristics of a known vibration source, namely a wooden staircase, a source used earlier in the earlier Mica2Dot Mote experiment [2]. A LabWorks ET-126 vibrating actuator driven by an Agilent 33120A signal generator and a LabWorks pa-138 power amplifier was used to emulate a vibration source which resonated at a frequency of 52 Hz at an acceleration magnitude of  $0.5 \text{ m/s}^2$ .



**Fig. 7.** “Teeny Temp” Piezoelectric Power Generator Mounted on a Small Vibration Table

A piezoelectric bimorph generator was designed to resonate at the same frequency. A piezoelectric bimorph from Piezo Systems with the dimensions  $31.5 \times 12.7 \times 0.51 \text{ mm}$  ( $1.25 \times 0.50 \times 0.02$ ”) was used in constructing the generator. This bimorph uses a brass center shim and PZT-5A4E. A 20g tungsten mass was affixed to one end using cyanoacrylate glue, and the device was mounted in a rigid plastic clamp. At the given acceleration magnitude, a 40V peak to peak AC signal resulted.

## 4 Regulator Circuit Results

Upon linking the various components of the “Teeny Temp” system, and experimenting, the design parameters of the power conditioning circuit were fine tuned. The super capacitor of  $22,000 \mu\text{F}$  was used with modified comparator trigger levels of 2.465V and 2.92V. These values resulted in a 2.1V output pulse for 3.32 seconds. Under automode interrogation, the reader was able to capture 15 reads from the tag per pulse. Time to charge the super capacitor from VLOW to VHIGH was 60 seconds.

An image of the notification message from the reader is shown in the figure below. The message shows tag ID, initial discovery time of the tag, and total number of reads. This particular notification message occurred after two pulses of power were supplied to tag. During those two pulses, 28 total reads occurred and the difference in time between the two pulses was 60 seconds.

Results for the Crossbow Mica2Dot Mote and the Alien RFID tag are compared in the table below. A fair comparison can be found in output power to load and time per cycle. In order to function properly, it was necessary to supply 68mW to the mote and 8mW to the RFID. Charge time for the power conditioning circuit for the mote application was 10 minutes and charge time for the power conditioning circuit for the RFID application was 1 minute.



```

#Alien RFID Reader Auto Notification Message
#ReaderName: Alien RFID Reader
#ReaderType: Alien RFID Tag Reader (Class BPT/2450Mhz)
#IPAddress: 0.0.0.0
#CommandPort: 23
#Time: 2005/04/29 13:45:28
#Reason: TIMED MESSAGE
Tag:1208 2004 1855 2000 0000 0000, CRC:1AEA, Disc:2005/04/29
13:44:29, Count:28, Ant:0
#End of Notification Message
SensorValue = 24

```

**Fig. 8.** Automated Notification from Alien RFID Reader under Automode

These results show a 88% decrease in output power and 90% decrease in charge time. There are two reasons for the decrease in charge time. First, the difference in trigger levels for the mote is around 1.5 V and the difference for the RFID is around 0.5 V. Therefore, a longer charge time is necessary to collect enough energy for the mote. Second, the trigger levels for the RFID are significantly lower than the trigger levels for the mote. Therefore, the storage capacitor on the RFID power circuit can more efficiently draw power from the piezoelectric power generator.

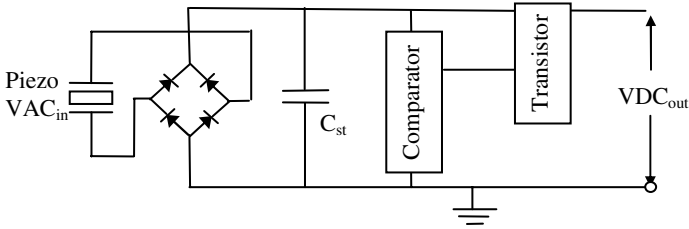
**Table 3.** RFID versus Mote Results

Characteristics	Mote	RFID
Capacitor Size	6,600 uF	22,000 uF
Trigger Levels	3.5 V → 5.4 V	2.47 V → 2.92 V
Energy Transferred	56 mJ	27 mJ
Bender Frequency and Accel	2.5 m/s <sup>2</sup> at 27 Hz	0.5 m/s <sup>2</sup> at 52 Hz
Bender Voltage Signal PP/RMS	40 VPP/ 14 VRMS	40 VPP/ 14 VRMS
Output Pulse to Load	3.3 V for 0.816 s	2.1 V for 3.32 s
Output Power to Load	68 mW	8 mW
Time per Cycle	10 min	1 min

## 5 A Better Power Conditioning Circuit

Upon analysis of the results in an effort to further optimize the “Teeny Temp” System, it became apparent that a regulator was not necessary and was only wasting usable energy. The range of voltage levels from the storage capacitor were all usable voltage levels of the RFID tag. There was no need to regulate that signal and burn off energy. Therefore, a transistor was substituted in place of the regulator. A schematic of the power conditioning is shown below.

Results from the modified circuit are shown in the graphs below. The top signal on each graph shows the voltage level of the capacitor while the bottom signal shows output voltage to the RFID tag. The new transistor circuit was analyzed under three different modes: no interrogation, manual interrogation, and automode interrogation. Under manual interrogation, the tags are called using a command prompt.

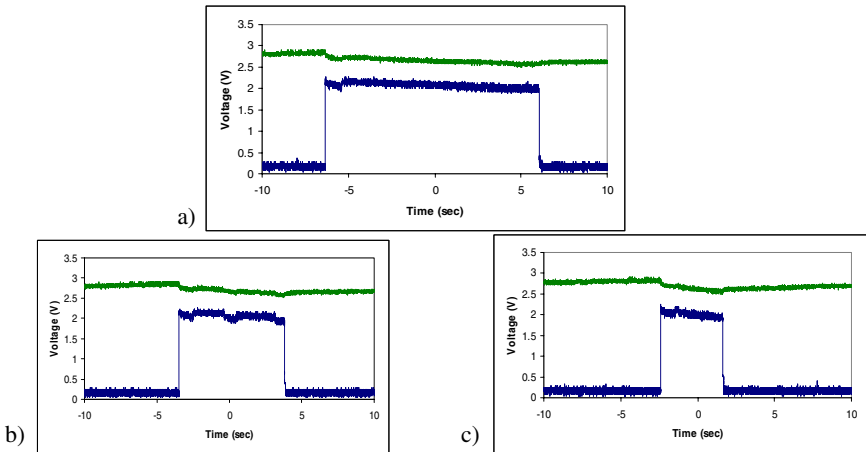


**Fig. 9.** Schematic of Power Conditioning Circuit with Transistor



**Fig. 10.** Power Conditioning Circuit with Transistor

There are several interesting things to observe from the graphs. First, there is a voltage drop between the capacitor and voltage output to the tag and this voltage drop occurs though the transistor. Therefore, a “no voltage drop transistor” would be more optimal in future designs. Second, the shape of the voltage output signal shows the benefits of a transistor versus regulator. Output from a regulator would be a flat line whereas the transistor allows the raw voltage to feed through. Therefore the energy saved can be represented by the area of the triangle which would otherwise have been burned to ground by the regulator.



**Fig. 11.** Capacitor Voltage (*top*) and Voltage Output to Load (*bottom*): a) No Interrogation b) Manual Interrogation c) Automode Interrogation

Third, the current draw during transmission can be found by comparing the pulse lengths between the different modes of interrogation. Under no interrogation, a pulse of 12.8 seconds is observed. The only current draw occurs from onboard sensor and electronics and is calculated to be 1.1mA. Under manual interrogation, where the tag was called twice, the length of pulse dropped to 7.12 seconds, suggesting an average current draw of 1.9mA under two manual interrogations. During automode interrogation, the pulse length drops to 4.2 seconds, suggesting an average current draw of 3.2mA.

**Table 4.** Regulator Circuit versus Transistor Circuit Results

Characteristics		Regulator Circuit	Transistor Circuit
Automode	Length of Pulse	2.9 sec	4.2 sec
	# of Reads	20	95
	Current Draw	4.4 mA	3.2 mA
Manual Interrogation	Length of Pulse	4.9 sec	7.12 sec
	# of Reads	19	8
	Current Draw	2.6 mA	1.9 mA
No Interrogation	Length of Pulse	9 sec	12.8 sec
	# of Reads	--	--
	Current Draw	1.4 mA	1.1 mA
Time to Charge		41 sec	40 sec
Output Voltage		2.1V flat	2.12 V to 1.88 V

The transistor circuit resulted in an average 44% increase in pulse length. A longer pulse length leads to more reliable transmissions to the reader. Or if shorter time to charge is desired, lower capacitance or lower trigger levels could be utilized since excess energy is no longer burned away via the regulator.

## 6 Conclusions

An experiment was conducted to evaluate the potential integration of energy scavenging from vibrations with battery-assisted passive RFID tags. The system chosen for analysis was the Alien 2.45 GHz battery-assisted passive tag system. The tags are equipped with onboard temperature sensor. Through experimentation, the tag was found to require at least 2.8mW over a pulse length of at least 2.5 seconds to reliably transmit information to the reader. A piezoelectric power generator was designed with a resonant frequency of 52 Hertz, with potential power output of 500  $\mu$ W [3,4]. A power conditioning circuit was designed to most efficiently convert AC power from the piezoelectric power generator to DC power to the RFID tag.

A 0.5m/s<sup>2</sup> acceleration magnitude vibration at 52 Hertz was emulated by a vibrating actuator. These vibrations in combination with the piezoelectric power generator and power conditioning circuit resulted in a 2.1 V signal for a pulse length of 3.32 seconds to the RFID tag. These values are equivalent to an 8mW power transfer to the tag. This value is in contrast to the 68mW required to power a Crossbow Mica2Dot Mote. Charge time for the power conditioning circuit for the mote applica-

tion was 10 minutes and charge time for the power conditioning circuit for the RFID application was 1 minute. Comparison of results between a battery-assisted passive RFID tag and Crossbow Mica2Dot Mote show a 88% decrease in necessary power supply and 90% decrease in charge time.

Upon further analysis, the power conditioning circuit was modified to optimize power transfer to the RFID tag. By replacing the regulator with a transistor, raw voltage coming from the capacitor was transmitted to the RFID tag since all voltage levels VLOW through VHIGH were acceptable by the RFID tag. Using the transistor resulted in a 44% increase in pulse length.

In summary, battery-assisted passive RFID tags with onboard sensor, are a feasible wireless sensor node solution for integration with energy scavenging from vibrations. Future work should focus on power needs of the tag with respect to distance from the reader. Since these tags rely on backscatter technology, it is assumed more power will be needed, the further away the tag is from the reader. Further optimization of the power conditioning circuit is also necessary. Transistors with no voltage drop should be utilized. In addition, capacitance of storage capacitor and trigger levels of comparator can be further optimized for specific applications. The system design engineer should acquire the necessary duty cycle for an application, then program circuit characteristics to operate at that duty cycle.

## References

1. Agarwal, V.: Assessing the Benefits of Auto-ID Technology in the Consumer Goods Industry. Auto-ID Centre (2001)
2. Leland, E., Lai, E., Wright, P.K.: A Self-Powered Wireless Sensor for Indoor Environmental Monitoring. WNCG Conference, Austin, TX (2004)
3. Roundy, S., Wright, P.K.: Energy Scavenging for Wireless Sensor Networks with Special Focus on Vibrations. Kluwer Academic Publishers. New York (2004)
4. Schmidt, V.H.: Theoretical Electrical power Output per Unit Volume of PVF<sub>2</sub> and Mechanical-to-Electrical Conversion Efficiency as Functions of Frequency. IEEE International Symposium on Application of Ferroelectrics, 6<sup>th</sup>, Bethlehem, PA (1986) 538-542

# Wireless RFID Networks for Real-Time Customer Relationship Management

Philipp Schloter<sup>1</sup> and Hamid Aghajan<sup>2</sup>

<sup>1</sup> Detecon, Inc., San Mateo, CA 94402  
pschloter@stanfordalumni.org

<sup>2</sup> Wireless Sensor Networks Lab, Department of Electrical Engineering,  
Stanford University, Stanford, CA 94305  
aghajan@stanford.edu

**Abstract.** A new system for real-time customer relationship management is proposed. The system is based on deploying a network of RFID readers throughout an environment. Information about the presence or lingering of participating customers at different times of day is collected providing valuable marketing information for better service provision. The implementation of the proposed system includes a database management program and an intuitive user interface allowing real-time access to the data acquired by the network.

## 1 Introduction

Increased competition is forcing retailers to leverage all information available in order to optimize their operations and improve their bottom-line. As part of this effort, understanding customers and managing the customer relationship is critical. As a result, the market for customer relationship management (CRM) [7, 8] software has exploded over the last decade – The Economist estimated the market for customer relationship management software alone at over \$11B for 2004 [1]. The next wave of growth in this area is already on the horizon – RT-CRM (real-time customer relationship management) [9]. RT-CRM will enable interacting with customers effectively around the clock. A major component of such an approach is tracking customer behaviour on retail and exhibition floors, which enables managers to optimize floor layout, and plan promotions and other marketing efforts accordingly. While existing prototypes are extremely expensive and complicated to deploy, our new system can offer similar capabilities at much lower cost. With the advent of decentralized wireless sensor networks [2], the cost of deploying sensing technologies will go down significantly. In this paper, we describe a novel prototype system of a wireless sensor network-based RT-CRM system, employing RFID [5] for sensing. The designed infrastructure lays the foundation for future research in the area of wireless sensor networks and next-generation CRM. By cutting the total system cost by at least an estimated factor of ten, wireless sensor networks will bring RT-CRM to the masses.

Existing system designs require a complete redesign of the floor space and hence are extremely expensive to deploy, limiting applicability to prototype environments or very large scale operations. An example for such a prototype environment is the Metro future store [3]. Metro designed a new store to demonstrate and test upcoming tech-

nologies relevant to retailers. An RFID-based shelf and shopping cart content tracking system was deployed at the Metro future store. RFID allows Metro to track its entire inventory electronically and enables customers to self check-out without a cashier.

Harrah's casino is an example of a very large scale non-prototype RT-CRM operation. Harrah's Entertainment casinos [6] introduced a customer loyalty card. Patrons need to insert their loyalty card into the casino's slot machines in order to play. Harrah's has over 36,000 such slot machines. The loyalty card uniquely identifies the patron and enables Harrah's to visualize the casino floor in real-time by attributes such as age, average spent, game or drink preferences. Harrah's can then optimize its casino floor [4] and even run custom-tailored promotions, such as offering the player her favorite drink, keeping her happy and playing – all in real-time.

The high cost of deploying and maintaining the type of systems as deployed by Harrah's or Metro limits the use of such real-time CRM technology to a select few, very large players in industry. While this type of technology can greatly improve revenue, it is prohibitively expensive. It requires a large data warehouse with specialized software and a complete redesign of the floor space to install the necessary electronics. Further, current technology cannot track customer movements on the floor unless customers insert or swipe a loyalty membership card at all places as they move, e.g. at slot machines. These constraints further limit the applicability of RT-CRM.

## 2 Approach

We propose a new low-cost RT-CRM system based on a wireless sensor network. In contrast to previous systems, our system is significantly less expensive to deploy. It requires neither extensive remodeling of the floor space nor computer specialists to deploy and operate the system. The system is almost completely self-configuring, allows tracking of customer movements even without swiping a loyalty card, and can be deployed without any redesign of the floor space. The system is applicable in a variety of environments, including retail, exhibitions and museums. We believe that our design will make RT-CRM a reality even for small- and medium-sized businesses.

### 2.1 Infrastructure

Our infrastructure consists of six components:

1. A set of wireless sensors, equipped with RFID readers, which collect readings from RFID-based loyalty cards.
2. An aggregator mote that accumulates the data from the sensors.
3. An aggregation server that receives the data from the aggregator mote and stores it into a database.
4. A database server that runs databases for a) sensor data, b) customer information and c) system settings. It also provides a hook to other back-end systems such as inventory or cash register management systems.
5. A data processing entity that mines the data in the database. The web server triggers this data processor.
6. A web server that hosts a web-based system administration interface and also visualizes the data results received from the data processor.

Graphically, the systems can be described as in Fig. 1.

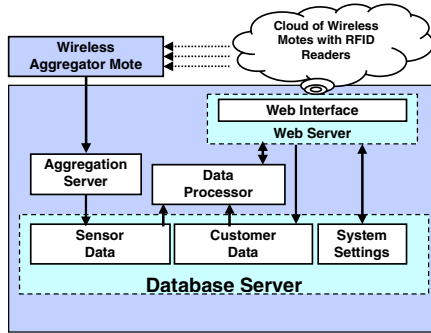


Fig. 1. Schematic overview of system infrastructure. Arrows indicate direction of data flow.

### 3 Implementation

In contrast to existing designs, our design does not require an entire redesign or upgrade of the floor space. In this section, we explain the deployment process in a typical retail environment.

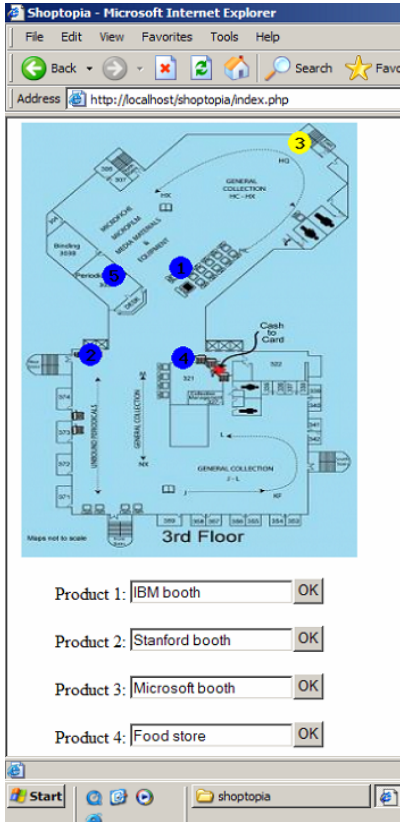
Instead of having to redesign the entire store to incorporate RT-CRM technology, a store manager simply walks through the store and places wireless sensor motes on the different shelves to track customers. This store “walk-through” is a one-time process. At the entrance, RFID-enabled loyalty cards are distributed to customers.

These cards are just the size of a credit card and carry a unique customer identifier that links to a customer entry in a database (they might also carry part or all of the actual customer data). The cards are wireless, so as a customer passes by a wireless sensor mote, the customer is automatically tracked. During checkout, an electronic cashier tracks the customer again and the information about the purchase is linked to the customer database entry. The store manager can add additional motes at anytime, increasing precision on the fly as required and reducing the initial investment.

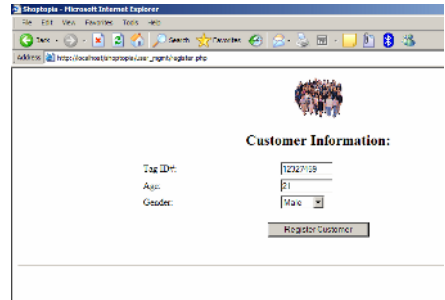
The store manager administers the system through a web-site. After having distributed the motes throughout the store, the store manager can load a map of the store via the web interface and graphically align the motes on the map (see Fig. 2(a)). This way the store manager knows exactly what store areas a set of customers visited.

#### 3.1 Operation and Analysis

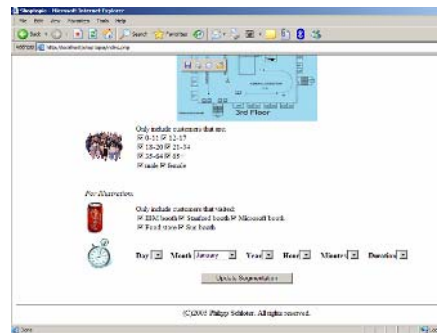
In this section, we describe our system’s easy-to-use maintenance and analysis web interface. Products can be added at any time and the corresponding sensors are added to the map. One could employ a system extension that even automatically positions the sensors on the floor map, e.g. via RSSI-based triangulation or other localization techniques. Such a feature would especially help for large deployments with many sensors. Since products can be labeled and added at any time, managers can grow the system organically. For example, initially only aisles might get distinguished, but then, as additional precision is required and budget allows, additional sensors are added, e.g. to distinguish different sections of an aisles or even individual shelves.



(a)



(b)



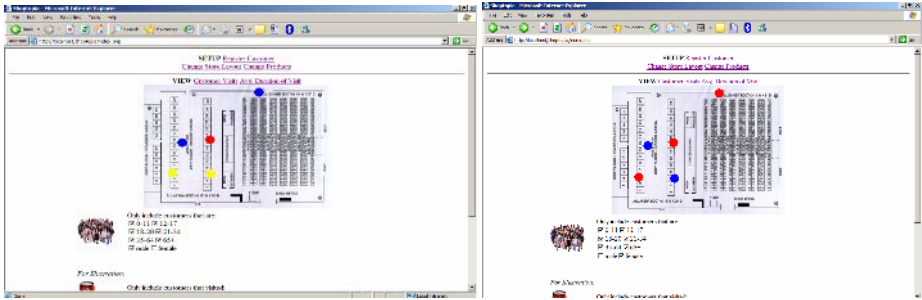
(c)

**Fig. 2.** Web interface (a) Arranging and labeling the sensors, (b) Entering customer information, (c) Example list of attributes for segmentation

In the prototype version, users sign up via the web interface. A stand-alone registration console is also possible. The current system requires users to enter their loyalty card number and their age and gender manually (see Fig. 2(b)) as the CRM data statistics. This step can be eliminated in a real deployment by enabling an RFID-based recognition of the card at the store front. This data is stored in the customer database and is used for segmentation. Clearly, the card number could also link to other customer information systems with more customer detail, e.g. ZIP code, level of education or customer’s purchase history. Such information systems are already employed in conjunction with store loyalty cards used for recording purchases.

The collected data is visualized via “heat maps.” The “# of visitors” view indicates the volume of customers. Red means the sensor’s visitor volume is in the top 25% percentile of all the sensors. Blue means the sensor’s visitor volume is in the bottom 25% percentile of sensors’ volumes. Yellow represents the 25% - 75% percentile, i.e. an average volume of visitors.





**Fig. 3.** Web interface – Example of difference in female and male in-store behaviours

Visitors can also be segmented by attributes such as age group, gender, products or booth visited. The data processing can be limited to a certain time span, e.g. early afternoon or between 1 and 2pm (see Fig. 2(c)). In addition, the average duration spent at a location is visualized. The average duration indicates how long customers take before they decide to purchase. The radius of each coloured circle indicates the range of the sensor. Red indicates long average duration, yellow indicates average duration (25% – 75% percentile) and blue indicates low average duration. Other visualizations are also feasible. For example, one could visualize the flow of customers between different locations. The mentioned segmentation functionally allows for example to illustrate how the in-store behaviour of female and male shoppers differs (see Fig. 3).

## 4 Data-Driven Decisions

Based on the visualizations, managers can quickly identify problem areas and deduce possible actions to improve performance, i.e. customer satisfaction and/or revenue.

For example, let's say the number of customers that visited a location is low and the percentage of those customers who bought the product carried at the location is low. No matter how much time customers spent at the location, it is likely that the target customer for the product simply rarely finds the product in the store. A manager should consider making the product more visible. For example, he could add signs to help direct customers or, if there is still no demand whatsoever for the product, the manager might decide to drop the product entirely.

If a lot of customers visited a location, spent a lot of time at the location, but did not buy after all, there might be too little information for the customers to decide on a purchase immediately. There could be many reasons. The store might carry too many similar products, there might be too little product information available for customers to form a decision, or it might just be the nature of the offering that purchase decisions take time. The manager should look into adding product information and/or investigate if adding sales staff to answer customer questions is feasible. Alternatively, there might also be problems with a product's packaging. This information could be valuable for the product management of the supplying company.

If a lot of customers visited a location, spent little time and bought little, it is likely that this passage is just a "walk-through," leading customers to other products. The manager might consider swapping out the product located here with a product that is

more relevant to the passing-by customers. This way the store could benefit from increased impulse purchases.

Many other scenarios can be deduced from the high-level visualizations provided by the system. Our system could even automatically detect certain data patterns and provide an according list of recommendations to the manager. The manager then could implement the recommended changes she thinks are most feasible. Finally, our system, linked to a cash register system, could measure the overall revenue impact and ROI of the implemented changes.

## 5 Cost Comparison

In this section, we will estimate the cost of deploying our wireless sensor network-based system and compare it with traditional systems. Since every deployment is different, it is challenging to provide an estimate for all cases. However, the following analysis should at least indicate the magnitude of potential cost savings.

**Table 1.** The major cost components in traditional systems and in the proposed technique.

	<b>Traditional</b>	<b>Proposed System</b>
<b>Software</b>	Highly specialized, costly software from multiple vendors – rarely web-based / rarely hosted service model.	Low-cost, single vendor and possibility to offer as hosted service with little up-front cost.
<b>Hardware</b>	Often times data center; multiple machines since different vendors might require different platforms and each application might require a dedicated machine by itself.	One PC (possibly on remote-side) ~\$1000 today Est. \$10/sensor (declining as adoption increases) Est. \$0.5/loyalty card (declining as adoption increases)
<b>Deployment</b>	Usually remodeling of floor space required; swap out of equipment (e.g. store shelves or slot machines), data cables need to be installed throughout the floor space.	Less than 30 minutes of initial training per manager Est. 30 seconds /sensor No data cabling to all aisles or shelves required.
<b>Maintenance</b>	Requires staff of IT experts.	Web-based, managed directly by manager. Products and sensors can be added any time.

The cost of a system can generally be divided into four areas: software, hardware, deployment, and maintenance. In Table 1, we list some of the key cost drivers in each area for traditional systems and for our system. Due to its high cost up to now, computer systems were only used by few entities in measuring and analyzing customer behaviour on the floor.

Traditionally, external or in-house consultants are hired sporadically to observe customers on the floor. By taking extensive notes, these consultants would analyze customer movement patterns, preferences and other behaviour to develop a set of recommendations on how to adjust the store layout. Hiring a consultant for a few hours can easily cost thousands of dollars and this traditional approach does not allow for the continuous tracking of trends. Continuous tracking is essential in a world of cyclical buying patterns, ever-changing product lineups, changing customer preferences and fast-moving competition. Considering the low margins in the retail business, a revenue increase due to better customer behaviour data by even a few cents per unit can yield significant additional profit. A mean of continuously tracking customer behaviour such as the proposed system can have a big business impact.

In Table 2, we estimate the cost of a traditional system vs. our new system. Please note that this calculation is only an estimate and that each scenario is likely to be different. However, these numbers should give an idea of the magnitude of the cost difference between the two systems. The required investment for the new system should be lower by a factor of 10 or more in comparison to traditional systems. Maintenance costs should be lower by a factor of 20 with our new design. Note that the fixed cost of the proposed system is significantly lower.

By utilizing a hosted service model with sensor network motes, the system can organically grow, reducing fixed costs and making it especially attractive to smaller stores. Clearly, the reduced cost will bring RT-CRM type applications to a broader audience. In addition, with the new system, businesses could start with a small deployment and add precision and sensors as needed, further lowering the required initial investment. Larger stores can benefit from the flexibility of the system by rotating the motes between different aisles, reducing the investment, while still producing an insightful view on customer behaviour.

**Table 2.** The costs are likely to be considerably less in the proposed system

	<b>Traditional</b>	<b>Proposed System</b>
<b>Software and Hardware</b>	~\$50,000	~\$1,500
<b>Sensors</b>	NA	~\$10,000
<b>Remodelling of shelves , Cabling</b>	~\$100,000	NA
<b>Deployment Labour</b>	~\$50,000	~\$1,000
<b>Maintenance Labour</b>	~\$100,000/year	~\$5,000/year
<b>Total</b>	~\$200,000 + \$100,000/yr	~\$12,500 + \$5,000/yr

## 6 System Integration Components

The first prototype system was implemented using a low-frequency RFID reader from Texas Instruments. The TI Passive RFID SDK kit was used for RFID reading. The server ran on MySQL and Apache on a laptop computer. Code was developed in various C derivatives as well as PHP. We are investigating several potential extensions to our initial prototype. Having the described infrastructure in place, new ideas and concepts can be quickly prototyped and tested. In particular, we have identified six key areas for extending the application:

1. **Sensing:** Includes long-range passive RFID technology, potentially aided by multiple reader antennas, simultaneous reading of very high volumes of RFID tags, and the automatic detection of possible intruders or hackers.
2. **Communication:** Includes finding the ideal trade-off point between aggregating data over a period of time at the mote sensor and the ability to have real-time information via high-frequency updates for different environments. Another area of concern is security, since the transmitted information might be of high interest for competitors.
3. **Storage:** Includes evaluation of the trade-off between only storing the required metrics for analysis, or storing all received data, which requires more storage. In addition, scalability and lifetime tests in real environments should be conducted.
4. **Visualization:** In this module the most relevant and actionable metrics are identified via conducting a user testing. In addition, real-time visualization and animation enhance the functionality of the system.
5. **Analysis:** Includes automatic detection of data patterns and events, and forming a list of appropriate recommendations. The automatic tracking of promotions and time series analysis in real-time enables another entire application.
6. **Administration:** Includes new ways of simplifying deployment and maintenance of the network. For example, automated localization of sensors upon deployment would eliminate the manual step of aligning the sensors with the floor map and, as a result, would greatly reduce the network setup time.

## 7 Customer Privacy

In order to facilitate adoption among customers of retail stores, the privacy concerns implied by the use of the tagged loyalty cards need to be addressed.

To this end, participating customers can be assured that the collected data is solely used for statistical analysis and no personally identifiable information is used in the market analysis process. Use of loyalty cards in super markets and wholesale membership clubs has proven successful in retaining participants via offering discount incentives. The user can be given an option to opt out from the CRM system, but discount offers would provide adequate incentives for many to participate.

In other settings, tags could be installed on shopping carts instead of having customers carry them. While this approach may provide an alternative to addressing privacy concerns, the disadvantage in adopting such method would be that customer segmentation may not be readily possible. In order to understand the behaviour of various different customer segments, it is essential that the proposed CRM system can link information with customer profiles.

Customers are willing to give up some privacy, if they gain something for it in turn. Client recognition service programs can provide a customized shopping experience. Customers gain by benefiting from customized services that are rendered to participants depending on the context and their stated preferences. Several casinos and membership-based recreation clubs have successfully deployed such customer service programs.

Finally, a single loyalty card that works across stores could be another option. Such a card can significantly improve service rendition to patrons. For example, the system could automatically guide the participating customers throughout a mall to the stores that carry their desired items.

## 7 Conclusions

The next wave of growth in customer relationship management is on the horizon – RT-CRM. In the future, firms will track customer behaviour on retail and exhibition floors and react in real-time by altering business levers such as floor layout, promotions, product mix and advertising. While existing prototype systems are extremely expensive and complicated to deploy, the system we implemented offers similar capabilities at much lower cost. By cutting the total system cost by at least an estimated factor of 10, wireless sensor networks will bring RT-CRM to the masses. Our prototype system lays the foundation for a plethora of new projects and will help to open up new fields, bridging customer relationship management and wireless sensor networks.

## Acknowledgments

We would like to thank the entire team at the Stanford Wireless Sensor Network Lab. We also would like to thank John Fogelsong and Texas Instruments for providing the RFID equipment.

## References

- [1] The Economist, March 2004 Edition, p. 95.
- [2] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. *A Survey on Sensor Networks*. IEEE Communication Magazine, August, 2002.
- [3] The Metro Store of the Future. <http://www.future-store.org/>
- [4] Compudigm Company Brochure, seePOWER™ V4 Gaming In Action
- [5] K. Finkenzeller. *RFID Handbook*. John Wiley & Sons. 1999.
- [6] David O. Becker. *Gambling on customers*. The McKinsey Quarterly, 2003 Number 2.
- [7] C.K. Prahalad, Patrica B. Ramaswamy, Jon R. Katzenbach, Chris Lederer, Sam Hill. *Harvard Business Review on Customer Relationship Management*. Harvard Business School Press, January 15, 2002.
- [8] P. Child, R. Dennis, T. Gokey, T. McGuire, M. Sherman, and M. Singer. *Can marketing regain the personal touch?* The McKinsey Quarterly, 1995 Number 3.
- [9] R. Sedgewick. *Real-time CRM: a competitive advantage today, a competitive imperative tomorrow?* Customer Interaction Solution, Volume: 22 Issue: 8 Page: 50(3), February 1, 2004.

# Tree-Based Classification Algorithm for Heterogeneous Unique Item ID Schemes

Yong Hwan Lee<sup>1</sup>, Hee Jung Kim<sup>1</sup>, Byeong-hee Roh<sup>1</sup>, S.W. Yoo<sup>1</sup>, and Y.C. Oh<sup>2</sup>

<sup>1</sup> Graduate School of Information and Communication, Ajou University,  
San 5 Wonchon-dong, Youngtong-Gu, Suwon, 443-749, Korea  
{lnj486, r1a81, bhroh, swyoo}@ajou.ac.kr

<sup>2</sup> Samsung Electronics Cooperation, Suwon, Korea  
ycoh@samsung.com

**Abstract.** For RFID-based applications, the uniqueness of ID assigned to each RFID tag should be guaranteed. Several research/standard organizations such as EPCglobal, ISO/IEC, Ubiquitous ID Center, and so on, have developed their own Unique Item ID (UII) specifications. The existence of various UII schemes may cause interoperability problems between applications using different UII schemes when those applications are operated on future global Internet network environment. In addition, it is expected that the traffic for UII query will be increased ten-times higher than that for DNS query in the current Internet. In order to overcome these problems, this paper proposes a fast tree-based classification algorithm applicable for various UII schemes, which can make it efficient to construct global directory lookup services for RFID applications with various UII schemes. Since the proposed scheme can be operated on readers, it can not only distribute traffic loads for UII queries, but also global RFID networks.

## 1 Introduction

Radio Frequency Identification (RFID) is a technology for the applications based on identification of the physical objects used for tracking and finding locations of objects. For RFID-related applications and services, the uniqueness of the identification assigned to each object should be guaranteed. Unique Item ID (UII) is an identification that uniquely identifies a specific object during its life.

There have been several standardization works for RFID UII by large research organizations. MIT Auto-ID Center developed a UII scheme called Electronic Product Code (EPC)[1], which is evolved from barcode systems widely used in supply-chains and management of manufactures. There are two types of UII schemes with ID lengths of 64-bit and 96-bit, respectively. Ubiquitous ID Center in Japan also developed its own UII scheme, called Ubiquitous Code (ucode)[2]. The ucode is the 128-bit long and can be extended as needed in 128-bit units such as 256, 384, or 512-bit. The ucode is a meta-code system so that it can be used to various kind of identification-based systems and services applied to intangible things such as services, softwares and so on, as well as materials made

by manufactures. ISO/IEC also developed ISO/IEC15459 series as UII schemes for RFID[3][4][5]. In ISO/IEC's UII scheme, each ID is represented by one to four numbers or characters using ASCII values[6]. Under the existence of various UII schemes currently as above, another works for developing new UII schemes such as [7] are being carried out. Likewise, it is expected that there will be various and complicated UII schemes.

The existence of various UII schemes may cause interoperability problems between applications using different UII schemes when those applications are operated on future global Internet network environment. Under global network environments, all UII query operations are done in public network infrastructures such as the current or future Internet, not in intra or closed network domains. The procedure for UII query is as similar as DNS query in the Internet. However, it is expected that the traffic for UII query will be increased ten-times higher than that for DNS. Under the circumstances, lookup speed for UII query can be a primary factor for the success of RFID-related services. EPCgloagl developed Object Name Service (ONS) for providing a global lookup service to translate an EPC UII into one or more Internet URLs which often identify an EPC information service [8]. However, the lookup service of ONS can support EPC's UII scheme only. In order to support global UII lookup services independent of any specific UII schemes, Multi-code Directory Service(MDS) has been developed by National Internet Development Agency(NIDA) of Korea [9]. MDS has been being operated as a trial system for RFID-based applications and services under public Internet environment in Korea since January 2005. To support global directory service for various UII schemes, MDS uses its own classification algorithm. The algorithm, for each UII query request, classifies what kind of UII scheme is necessary for the resolution of the request. Then, MDS sends the request to appropriate resolving server. However, since the classification algorithm used in MDS is done according to a given priority to a specific UII scheme such as EPC first, its performance decreases as the occupancy ratio specific UII scheme decreases. In addition, all classification processes for UII queries are done in MDS, the burden for operating the classification algorithm in MDS increases as those query requests increase.

This paper proposes a fast and efficient tree-based classification algorithm of UII schemes for global UII directory lookup services. As in MDS, the proposed algorithm is used to determine the appropriate UII scheme necessary for resolving each UII query request. It is noted that unlike DNS query request, RFID tags can provide only ID information consisting of binary bits. Since RFID applications need to know some additional information related to each tag ID, they have to ask it to UII resolving server such as EPCIS (EPC Information Service). However, with tag ID itself, it can not be known what UII resolving server it can ask to under the environments where various UII schemes are working together. The proposed algorithm can solve the problem with the following features. First, it guarantees a constant performance regardless of the types of UII schemes and their distributions, unlike the algorithm used in MDS whose performance depends on the penetration ratio of specific UII scheme. Second, it

can be operated at each reader, it can not only decentralize traffic loads for UII queries, but also make it easy to manage global RFID networks.

The paper is organized as follows. In Section 2, we briefly explain some related works such as existing UII schemes and the classification algorithm used in MDS. The proposed algorithm and its performance results are presented in Section 3 and Section 4, respectively. Finally, we conclude the paper in Section 5.

## 2 Related Works

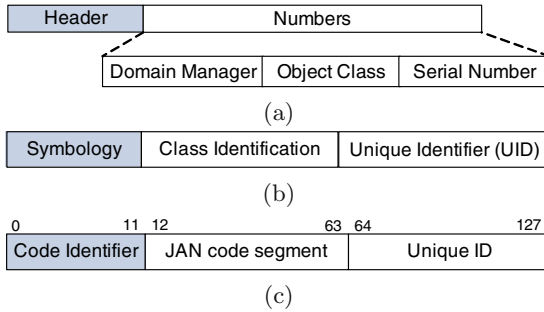
### 2.1 Unique Item ID (UII)

**EPCglobal.** The general structure of the Electronic Product Code (EPC) consists of a fixed length header and a series of numbers, as illustrated in Fig.1(a)[1]. The Numbers field comprises three entities such as Domain Manager, Object Class, and Serial Number. The substructure of the entities of the Numbers field and its own pertinent role are determined by the header value. The Domain Manager identifies the EPC manager of the UII. This field is assigned to institutions or companies responsible for maintaining the Object Class and Serial Number. The Object Class indicates the group or class of the object, and the Domain Manager must guarantee unique assignment of Object Class numbers within its own managing domain. The Serial Number is a specific number for every products and should be uniquely assigned within the range of the Object Class.

**ISO/IEC.** In Fig.1(b), the general UII structure defined by ISO/IEC 15459 is shown [3][4][5]. In SI (Symbology Identifier), the type of data format defined by AIM (association for Automatic Identification and Mobility) is written. SI can be used for decoding the data expression of UII. It is noted that SI is not included in RFID tag ID. After reader reads the tag ID, SI is optionally added into the UII by the reader. The Class Identification field indicates the class of the UID (Unique Identifier). The class information is used to identify products according to the specific rule defined for each class. The UID is an identifier uniquely assigned to every individual product by the issuing agency.

**Ubiquitous ID Center.** The ucode (ubiquitous Code) is an UII scheme developed by the ubiquitous ID center in Japan. The length of the ucode is basically 128-bit long, and can be extended as needed in 128-bit units such as 256-bit, 384 or 512-bit [9]. Since the ucode adopts the meta-code system, it can be used to various kind of identification-based systems and services applied to intangible things such as services, softwares and so on, as well as materials made by manufactures. The 128-bit length of ucode can accommodate ISBN (International Standard Bibliographic Number), ISSN (International Standard Serial Number) publication ID, and existing barcode scheme. And, it can accommodate IPv6 addresses. An example of ucode structure is shown in Fig.1(c), in which it can accommodate all barcodes used in Japan by encoding the JAN (Japan Article Number) code. Code Identifier indicates the type of code used in the ucode. In





**Fig. 1.** Various UUI schemes (a) EPC UUI format example (b) general UUI structure of ISO/IEC (c) ucode example

the example, Code Identifier indicates that the following code is used for the JAN code. As in the example shown in Fig.1(c), where JAN code is used, Unique ID with 64-bit long identifies individual items uniquely distinguished from others.

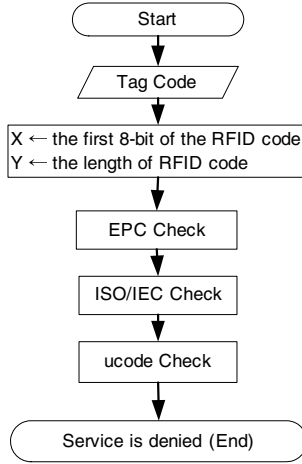
## 2.2 Multi-code Directory Service(MDS)

As mentioned above, the existence of various UUI schemes and the expected rapid growth of UUI query traffic in future global RFID network environments may cause severe problems in managing and deploying RFID services and networks. Under such circumstances with heterogeneous UUI schemes, fast global lookup for UUI query can be a primary factor for the success of RFID-related services.

**MDS.** MDS[9] has been developed for the global RFID directory services to provide interoperability between applications and services with different UUI schemes under the public Internet environment, especially, EPC, ISO/IEC and ucode. MDS has been being operated as a trial system in Korea since January 2005. The purpose of MDS is to implement global RFID network environment independent of UUI schemes.

**Classification Algorithm Used in MDS.** RFID tags can provide only ID information consisting of binary bits. Since RFID applications need to know some additional information related to each tag ID, they have to ask it to UUI resolving server such as EPC's ONS. However, with tag ID itself, it is impossible to know what UUI resolving server they can ask to under the environments where various UUI schemes are using together. MDS solves the problem as follows. All readers send UUI query messages to MDS. Then, the classification module used in MDS figures out what UUI scheme is used in the tag ID. According to the result from the classification module, MDS send the query message to corresponding UUI resolving servers in behalf of readers. Then, the resolving message can be sent back to readers through MDS.

Fig.2 shows the procedure of the classification algorithm used in MDS. In MDS, it is assumed that the standardization processes in EPCglobal and ISO/



**Fig. 2.** Basic process of classification algorithm used in MDS

IEC are faster than others, and these UII schemes of two groups will be used on commercial business earlier. From the expectation, as shown in Fig.2, the classification algorithm used in MDS is done according to a given priority to a specific UII scheme such as EPC first, ISO/IEC next, and finally ucode. Accordingly, the algorithm has some weakness that its performance decreases as the occupancy ratio of specific UII scheme decreases. In addition, since all classification processes for UII queries are done in MDS, the burden for operating the classification algorithm in MDS increases as those query requests increase.

### 3 Tree-Based Classification Algorithm

In this Section, we explain our proposed tree-based UII classification algorithm. As shown in Fig.3, we can make a binary tree for classifying between EPC and ISO/IEC’s UII schemes according to the specifications of them. When it is necessary for a new UII scheme to be added, we can easily add the new scheme into the tree as in normal binary tree manipulation process. In the tree shown in Fig.3, except for the only case when initial two bits of header are ‘10’, there is no confusion to classify between UII schemes. That is, following to the binary tree, these UII schemes can be classified simply when initial two initial bits are not ‘10’. In case when two initial bits are ‘10’, it needs additional process to classify between the two schemes, SGTIN-64 of EPC and Alphabet UII of ISO/IEC. In this case, by comparing the length and the consistency in the fields of the UII as illustrated in Section 2, the two schemes can be easily classified.

To speed up the classification process, instead of comparing all bits as in MDS, only necessary minimum bits are used for the classification. Except for ISO/IEC’s UII and EPC’s SGTIN-64 UII scheme, other EPC’s UII schemes are



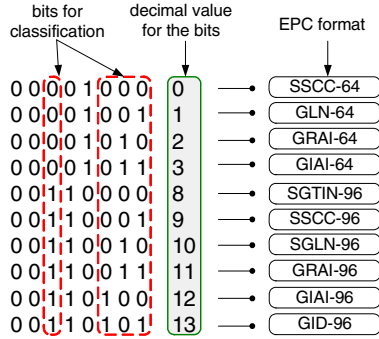


Fig. 4. Bits for classifying EPC’s UII schemes

Table 1. Comparisons of algorithm used for the experiment

Algorithm	MDS-A	MDS-C	PROPOSED
Process	Priority-based	Priority-based	Bit information
Initial assumption	8-bit header/length	8-bit header/length	nothing
Procedure(EPC)	1.header(EPC-64) 2.length	1.length(EPC-96) 2.header	Bit sequence
Procedure(ISO)	Check if the header includes numbers or characters in leading 8-bit		Bit sequence
Error handling	Out of consideration		

plexity. Accordingly, it can not only decentralize traffic loads for UII queries, but also make it easy to manage global RFID networks.

### 4 Experimental Results

For the experiments, UIIs are generated randomly according to EPC and ISO/IEC 15459 specifications. To show the efficiency of the proposed algorithm, we compared its performances with those of algorithms used in MDS. The features of those comparable schemes are shown in Table 1. In Table 1, whereas PROPOSED denotes our proposed algorithm, MDS-A and MDS-C are the classification algorithms used in MDS[10]. MDS-A and MDS-C have different priority strategies to classify UII schemes as shown in Table1. Especially, MDS-C is working in the MDS trial service operated by NIDA. In MDS-A and MDS-C, it is assumed that the two algorithms already know the 8-bit header information and the length of the UII before the classification[10]. However, the PROPOSED performs without the knowledge of them, and its classification process is done based on the bit-by-bit operation on tag IDs.

**Accuracy of Classification.** In order to check the accuracy of the algorithms, we made artificial UIIs according to corresponding schemes, and mixed them randomly. It is noted that there are only valid UIIs in the mixed sequences.

```

Algorithm FilteringBasedOnTree
1: begin
2:   if (the first bit of the Code is 'one')
3:     if (the length of the Code is more than 64) //ISO Code
4:       Operate ISO Code Process
5:     else if (the length of the Code is equal 64) //EPC Code
6:       Operate EPC Code Process
7:     end if
8:   else if (the second bit of the Code is 'one') //ISO Code
9:     Operate ISO Code Process
10:  else if (the bit is third, sixth, seventh or eighth bit)
11:    Save the bit value for EPC Code Process
12:    if (the bit is eighth bit) //EPC Code
13:      Operate EPC Code Process
14:    end if
15:  end if
16: end
    
```

Fig. 5. Pseudo code of the proposed algorithm

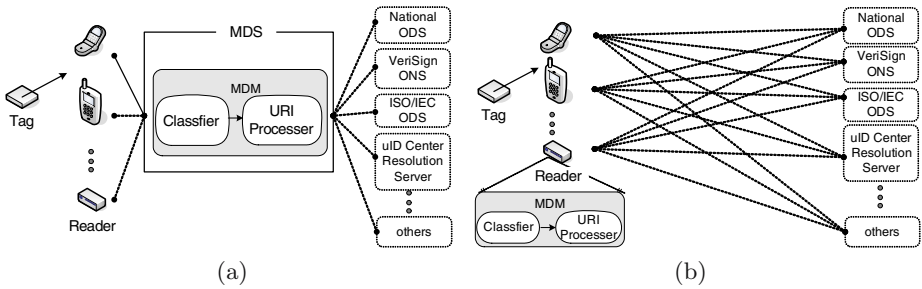
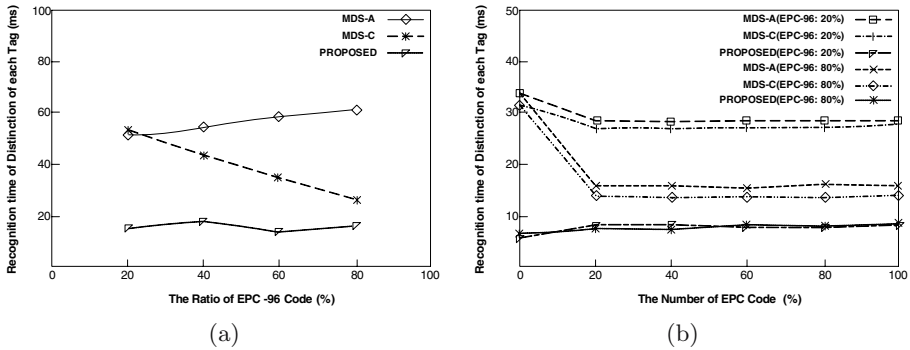


Fig. 6. Application architectures of (a) MDS and (b) proposed scheme

Then, we tested the classification algorithms using the mixed UII sequences, and found that there was no failure in classification by all the algorithms.

**Effect on Length of EPC UII.** As mentioned before, different UII schemes may have different code lengths. For example, the length of each EPC UII is either 64-bit (EPC-64) or 96-bit (EPC-96). In order to find out how the algorithms are affected by the various lengths of UII schemes, we carried out the following experiment. First, we randomly generated 10000 ISO/IEC UIIs and 10000 EPC UIIs varying the ratio between EPC-64 and EPC-96. Then, we mixed the UIIs randomly. Fig.7 shows the processing time for classifying all UIIs using each algorithm varying the occupancy ratio of the EPC-96 to EPC-64. Each algorithm was implemented by using C language, and we executed those algorithms on the computer of a Pentium-4 2.80GHz with Windows XP OS. We carried out 100 times of experiments, and its average values are shown in Fig.7(a). In the horizontal axis of Fig.7(a), the value 20% means there are 20% EPC-96



**Fig. 7.** Processing time varying (a) the ratio of EPC-96 (b) the occupancy ratio between UII schemes

UIIs out of total 10000 EPC UIIs while ISO/IEC’s UIIs are constant at 10000. As mentioned before, it is noted that MDS algorithms perform the classification with higher priority to EPC UIIs, but MDS-A handles the EPC-64 scheme first while MDS-C does the EPC-96 first. As the ratio of EPC-96 increases, the processing time of MDS-A increases also, while the processing time of MDS-C decreases. However, the proposed algorithm provides constant performances independent of the variance of the ratio of EPC-96. And, the proposed algorithm shows much lower processing time performances than MDS-A and MDS-C. It is expected that in future RFID network environments EPC-96 will be used much more than EPC-64. This experiment show that the proposed algorithm can be applied to both current and future global RFID network environments more efficiently than MDS algorithms.

**Effect on Occupancy Ratio of UII Schemes.** Under the existence of various UII schemes, the performance varying the occupancy ratio of UII schemes is one of the most important factors to compare the classification algorithms. In terms of performance, especially, if the performances of algorithms depend on the occupancy ratio of certain UII schemes, it may cause severe problems in managing RFID services and networks. In Fig.7(b), we show the performances of the comparable algorithms by varying the occupancy ratio of UII schemes. For obtaining Fig.7(b), we generated total 10000 UIIs. Among the 10000 UIIs, we varied the ratio of EPC UIIs from 0% where there are only ISO/IEC UIIs to 100% where no ISO/IEC UIIs exist. Among EPC UIIs, there are two cases when the ratios of EPC-96 are fixed at 20% and 80%. These two cases are reflecting the environments of RFID applications in the current and future RFID networks. In future RFID networks, it is expected that more EPC-96 UIIs are used than EPC-64. From Fig.7(b), the occupancy ratio between UII schemes can affect the performances of algorithms MDS-A and MDS-C. This phenomenon can be explained as same as in the case of Fig. 7(a). However, the proposed algorithm shows much lower processing time than MDS algorithms independent of the occupancy ratios of UII schemes.

## 5 Conclusion

In this paper, we proposed a fast and efficient tree-based algorithm for classifying various UII schemes in global UII directory lookup services. We show that the proposed algorithm can provide constant performances independent of UII schemes and lower processing time than the current MDS algorithms. And, the proposed algorithm can be operated at each reader, it can not only decentralize traffic loads for UII queries, but also make it easy to manage global RFID networks. Though we explained our proposed scheme in the case for two UII schemes such as EPC and ISO/IEC, it can be easily extended to all kind of UII schemes by adding new required UII schemes into the tree.

It is expected that the traffic for query will be increased ten-times higher than that for DNS. Under the circumstances, lookup speed for UII query can be a primary factor for the success of RFID-related services. The main features of the proposed algorithm such as faster speed, the independence of UII schemes and the operation at readers can be applied in global RFID network environments.

## Acknowledgement

This work was supported by grant (No. 05A3-I3-10) from Ubiquitous Autonomic Computing and Network Project sponsored by the Ministry of Information and Communication, Korea.

## References

1. EPCglobal, "EPC Tag Data Standard Version 1.1 Rev.1.26," Nov. 2004
2. Ken Sakamura, "Ubiquitous ID Center has authorized 2 types of RFID chips made by Fujitsu as the Standard ucode tag," uID Center, Dec. 2004
3. ISO/IEC 15459-1, "Information technology-Unique identifiers for item management Part 1: Unique identification of transport unit," Sep. 2004
4. ISO/IEC 15459-3, "Information technology-Unique identifiers for item management Part 3: Common rules for unique identification," Sep. 2004
5. ISO/IEC 15459-4, "Information technology-Unique identifiers for item management Part 4: Unique item identification for supply chain management," Sep. 2004
6. Material Handling Industry, "Data Identifier and Application Identifier Standard, Standard Under Continuous Maintenance," Jun. 2004
7. KISTI, "China, Global RFID standard development propulsion," Overseas scientific technical trend TSTA200510135367, Mar. 2005
8. Auto-ID Center, "Auto-ID Object Name Service (ONS) 1.0," Aug. 2003
9. NIDA, "MDS Guide Line V1.0," Guide Manual, Dec. 2004
10. S.W. Yoo, et al., "Registration and Management in RFID ONS and Policy Research," Final Report, NIDA, Nov. 2004

# An Efficient Key Pre-distribution Scheme for Secure Distributed Sensor Networks\*

Sung Jin Choi and Hee Yong Youn\*\*

School of Information and Communications Engineering,  
Sungkyunkwan University, Suwon, Korea  
{choisj, youn}@ece.skku.ac.kr

**Abstract.** Wide-spread deployment of sensor networks is emerging and it presents an economical solution to numerous problems. A number of applications are dependent on secure operation of the sensor network, however, and serious consequences are incurred if the network is compromised or disrupted. In the existing key pre-distribution scheme suitable for low power and resource sensor nodes, shared key is not guaranteed to be found and mutual authentication is not allowed. This paper thus proposes a new key pre-distribution scheme guaranteeing that any pair of nodes can find a common secret key between themselves by using the keys assigned by LU decomposition of a symmetric matrix of a pool of keys. Furthermore, it allows node-to-node mutual authentication. Analysis shows that the existing scheme requires a large number of keys in each sensor node to display a comparable performance as the proposed scheme. Therefore, the superiority of the proposed scheme is more substantial when the memory size of the sensor node is small.

**Keywords:** distributed sensor network, key pre-distribution, LU decomposition, mutual authentication, security.

## 1 Introduction

Wide-spread deployment of sensor networks is on the horizon. Networks of thousands of sensors may present an economical solution to some of the challenging problems: real-time traffic monitoring, monitoring of building safety (structural, fire, and physical security monitoring), military sensing and tracking, distributed measurement of seismic activity, real-time pollution monitoring, wild life monitoring, wild fire tracking, etc [1].

Distributed sensor networks (DSNs) share several characteristics with the traditional wireless networks. Both include arrays of sensor nodes that are battery powered, have limited computational capabilities and memory, and rely on intermittent wireless communication via radio frequency and, possibly, optical links. They also include data-collecting nodes which cache sensed data and make them available to the

---

\* This research is supported by the Ubiquitous Autonomic Computing and Network Project, 21st Century Frontier R&D Program in Korea and the Brain Korea 21 Project in 2005.

\*\*Corresponding author.



application components of the network for processing, and control nodes which monitor the status of sensor nodes and broadcast simple commands to them. However, DSNs differ from the traditional wireless networks in several aspects, namely: their scale is a few orders of magnitude larger than that of wireless networks; they are dynamic in the sense that they allow addition and deletion of sensor nodes after deployment to extend the network or replace failing or unreliable nodes without physical contact; and they may be deployed in hostile areas where communication is monitored and the sensor nodes are subject to capture and manipulation by an adversary. These challenging operational requirements place equally challenging security constraints on the DSN design [2,3].

Many applications are dependent on secure operation of the sensor network, and have serious consequences if the network is compromised or disrupted. Also, when the sensor networks are deployed in a hostile environment, security becomes extremely important as they are prone to different types of malicious attacks. For example, an enemy can easily tap the information, imitate one of the sensor network nodes, or intentionally provide fault information to other nodes [4]. The problem here is how to secure the communication between the sensor nodes, i.e. how to set up secret keys between communicating nodes. Most earlier schemes use asymmetric cryptography to solve this problem [10]. However, these schemes are often not suitable for distributed sensor network due to limited computation and energy power of the sensor nodes.

To address this issue a scheme has been recently proposed which is based on random key pre-distribution. However, it also has a shortcoming that a common key is not guaranteed to be found between two nodes wanting to communicate. This paper thus proposes a new key pre-distribution scheme which guarantees that any pair of nodes can find a secret key between themselves by using a pool of keys formed in the symmetric matrix format and the relevant property of LU decomposition of a matrix [13]. Furthermore, it allows node-to-node mutual authentication which the existing scheme does not support. Analysis shows that the existing scheme requires a large number of keys in each sensor node to display a comparable performance as the proposed scheme. Therefore, the superiority of the proposed scheme is more substantial when the memory size of the sensor node is small.

The rest of the paper is organized as follows. Section 2 discusses the existing key distribution approaches for sensor network, and Section 3 presents the proposed scheme. Section 4 analyzes and compares the performance of the proposed scheme with the earlier scheme, and finally concluding remark is given in Section 5.

## 2 Related Works

The traditional key exchange and distribution protocols based on the infrastructure of the internet using trusted third parties are impractical for large scale DSNs because of the network topology unknown prior to deployment, communication range limitation, intermittent sensor-node operation, and network dynamics, etc. To date, the only practical option for the distribution of keys to sensor nodes of large-scale DSNs whose physical topology is unknown prior to deployment would have to rely on key pre-distribution. Keys would have to be installed in the sensor nodes to accommodate secure connectivity between the nodes. However, the traditional key pre-distribution approach requires either a single mission key or a set of separate  $n-1$  keys, each being

privately shared with another node pair-wise, must be installed in every sensor node. This is an inadequate aspect for the DSNs [5].

There exist a number of key pre-distribution schemes. One solution is to let all the nodes carry a master secret key. Any pair of nodes can use this global master secret key to achieve key agreement and obtain a new pairwise key. This scheme does not exhibit desirable network resilience; if one node is compromised, the security of the entire sensor network will be compromised. Some existing studies suggest storing the master key in tamper-resistant hardware to reduce the risk [6], but this increases the cost and energy consumption of each sensor node. Furthermore, tamper-resistant hardware might not always be safe. Du et al. [7] proposed another key pre-distribution scheme which substantially improves the resilience of the network compared to other schemes. This scheme exhibits a threshold property; when the number of compromised nodes is smaller than the threshold, the probability that any node other than the compromised nodes is affected is close to zero. This desirable property lowers initial payoff of small scale network breaches to an adversary, and makes it necessary for the adversary to attack a significant portion of the network.

Blundo et al. [8] proposed several schemes which allow any group of some parties to compute a common key while being secure against collusion between some members of them. These schemes focus on saving communication cost while memory constraints are not placed on the group members. Perrig et al. [9] proposed SPINS, a security architecture specifically designed for sensor networks. In SPINS, each sensor node shares a secret key with the base station. Two sensor nodes cannot directly establish a secret key. However, they can use the base station as a trusted third party to set up a secret key.

Recently, Eschenauer and Gligor [10] proposed a random key pre-distribution scheme. Here a pool of random keys is selected from a key space. Each sensor node receives a subset of random keys from the pool before deployment. Any two nodes able to find one common key within their respective subsets can use it as their shared secret to initiate communication. Based on this scheme, Chan, Perrig, and Song [11] proposed a  $q$ -composite random key pre-distribution scheme, which increases the security of key setup such that an attacker has to compromise many more nodes to achieve a high probability of compromising communication. The difference between the  $q$ -composite scheme and the scheme in [10] is that  $q$  common keys ( $q \geq 1$ ), instead of just a single one, are needed to establish secure communication between a pair of nodes. It was shown that network resilience against node capture is improved by increasing the value of  $q$ . The main issues in the random key pre-distribution approach are that a common key may not be found between a pair of nodes and node-to-node mutual authentication is not allowed. We next present the proposed scheme solving these problems.

### 3 The Proposed Scheme

In this section we present the basic features of the proposed scheme, deferring its analysis to the next section. First, we briefly describe how the proposed key pre-distribution scheme works. The proposed scheme uses a random graph like the Eschenauer's method [10]. It, however, guarantees that any pair of nodes can find a secret key between themselves along with mutual authentication.

### 3.1 Preliminaries

We capitalize some important properties of matrix in designing the key pre-distribution scheme.

**Definition 1.** If a square matrix  $K$  has the property  $K^T = K$ , where transpose of matrix  $K$  is denoted by  $K^T$ , we say that  $K$  is a symmetric matrix. A symmetric matrix means that  $K_{ij} = K_{ji}$ , where  $K_{ij}$  is the element in the  $i$ th row and  $j$ th column of matrix  $K$ .

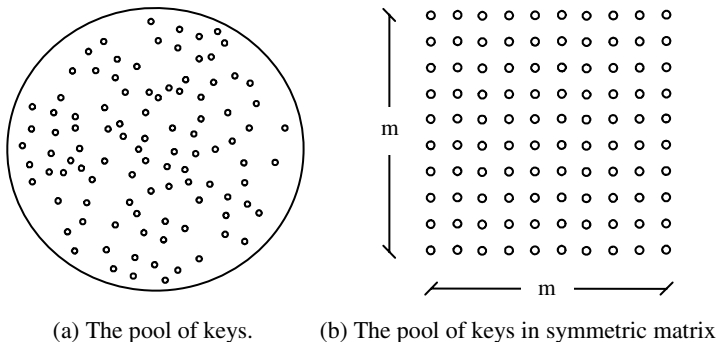
**Definition 2.** LU decomposition is to decompose an  $m \times m$  matrix  $K$  into two matrices such that  $K = LU$ , where  $L$  is an  $m \times m$  lower triangular and  $U$  is an  $m \times m$  upper triangular matrix, respectively, i.e., product of the lower triangular matrix  $L$  and upper triangular matrix  $U$  gives rise to  $K$ .

### 3.2 The Proposed Key Distribution Scheme

We now explain the proposed key pre-distribution scheme. The key pre-distribution scheme consists of four off-line steps; namely generation of a large pool of keys (e.g.,  $2^{17} \sim 2^{20}$  keys), forming a symmetric matrix using the pool of keys, applying LU decomposition to the symmetric matrix, and key pre-distribution to each sensor node. We discuss the four steps next.

**Step 1 (Generation of a large pool of keys (e.g.,  $2^{17} \sim 2^{20}$  keys)):** We propose a key pre-distribution scheme using the random key approach. In the proposed key pre-distribution scheme each sensor node receives a subset of random keys from a large pool of keys before deployment. For communication, two nodes need to find one common key to use it as their shared secret key. Therefore, the base station first needs to generate a large pool of keys (e.g.,  $2^{17} \sim 2^{20}$  keys) in this step.

**Step 2 (Forming a symmetric matrix using the pool of keys):** Eschenauer’s a random key pre-distribution scheme uses just a large pool of keys as shown in Figure 1(a). However, the proposed scheme uses a pool of keys formed in a symmetric matrix as shown in Figure 1(b).



**Fig. 1.** The pool of keys

**Step 3 (Applying LU decomposition to the symmetric matrix):** We apply LU decomposition to the symmetric matrix to let a pair of nodes always find a common key between themselves and raise the security by providing node-to-node mutual authentication.

**Step 4 (Key pre-distribution):** In this step every node is randomly assigned one row from the L matrix and one column from the U matrix, respectively. One and only one condition here is that the same row and column position are assigned such that  $L_{r_i}$  (ith row of L) and  $U_{c_i}$  (ith column of U) are assigned to each node.

**(Finding a common key):** Assume that node\_x and node\_y contains  $(L_{r_i}$  and  $U_{c_j})$  and  $(L_{r_j}$  and  $U_{c_i})$ , respectively. When node\_x and node\_y need to find a common secret key between them, they first exchange their columns, and then compute a vector product as follows.

$$\begin{aligned} \text{node\_x: } & L_{r_i} \times U_{c_j} = K_{ij} \\ \text{node\_y: } & L_{r_j} \times U_{c_i} = K_{ji} \end{aligned}$$

Recall that K is a symmetric matrix, and thus  $K_{ij} = K_{ji}$ .  $K_{ij}$  (or  $K_{ji}$ ) is then used as a common key between node\_x and node\_y. Note that the proposed scheme allows any pair of nodes to always find a common secret key between themselves.

**Example:** We illustrate the proposed scheme using an example below.

**Step 1:** We first generate a large pool of keys using a random graph in this step, and assume here that we generate a pool of keys, S (-5~5).

**Step 2:** After we select (-2, 1, 2, 4) from the pool of keys, S, arrange them into a symmetric matrix K.

$$K = \begin{bmatrix} 2 & 4 & -2 \\ 4 & 1 & 2 \\ -2 & 2 & 1 \end{bmatrix} : \text{The pool of keys in a symmetric matrix}$$

**Step 3:** We apply LU decomposition to the symmetric matrix. We first calculate the

elementary matrix  $E_1 = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ ,  $E_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$ , and  $E_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 6/7 & 1 \end{bmatrix}$  to

derive L and U. Then we calculate  $L = E_3 E_2 E_1 A$  and  $U = E_1^{-1} E_2^{-1} E_3^{-1}$ . As a result, L and U are obtained as follows.

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & -6/7 & 1 \end{bmatrix}, U = \begin{bmatrix} 2 & 4 & -2 \\ 0 & -7 & 6 \\ 0 & 0 & 29/7 \end{bmatrix}$$

**Step 4:** This step is for key pre-distribution, and assume that  $L_{r_3}$  and  $U_{c_3}$  are stored at node\_x. Similarly,  $L_{r_2}$  and  $U_{c_2}$  are stored at node\_y. When node\_x and node\_y need to find a secret key between them to securely communicate, they first exchange their columns, and then calculate the key value, respectively. Here the value turns out to be

2. Then they compare them for authentication. Since the values are same, they can authenticate each other and start communication using ‘2’ as the shared key. The process is summarized in Table 1.

**Table 1.** The operations for key authentication.

	node_x	node_y
After key pre-distribution	$L_{r_3} (-1, -6/7, 1)$ $U_{c_3} (-2, 6, 29/7)$	$L_{r_2} (2, 1, 0)$ $U_{c_2} (4, -7, 0)$
After column-exchange	$L_{r_3} (-1, -6/7, 1)$ $U_{c_2} (4, -7, 0)$	$L_{r_2} (2, 1, 0)$ $U_{c_3} (-2, 6, 29/7)$
After key-computation	$L_{r_3} \times U_{c_2} = 2 (=K_{32})$	$L_{r_2} \times U_{c_3} = 2 (=K_{23})$

### 3.3 Node-to-Node Mutual Authentication

The existing random key pre-distribution scheme does not allow node-to-node mutual authentication, but the proposed scheme based on the symmetric matrix of the keys does that as follows. Table 2 summarizes the process of node-to-node mutual authentication.

1. node\_x sends  $U_{c_i}$  (the column it contains) to node\_y.  
node\_x  $\rightarrow$  node\_y : {  $U_{c_i}$  }
2. node\_y: obtains  $K_{ji}$  by multiplying  $L_{r_j}$  with  $U_{c_i}$  received from node\_x, and then sends  $U_{c_j}$  and  $K_{ji}$  to node\_x.  
node\_y : {  $L_{r_j} \times U_{c_i} \rightarrow K_{ji}$  }  
node\_y  $\rightarrow$  node\_x : {  $U_{c_j}, K_{ji}$  }
3. node\_x: obtains  $K_{ij}$  by multiplying  $L_{r_i}$  with  $U_{c_j}$  received from node\_y, and compares it with  $K_{ji}$  received from node\_y.  
node\_x : {  $L_{r_i} \times U_{c_j} \rightarrow K_{ij}$ , check if  $K_{ij} = K_{ji}$  }
4. If node\_x verifies  $K_{ij} = K_{ji}$ , then sends  $K_{ij}$  to node\_y.  
node\_x  $\rightarrow$  node\_y : {  $K_{ij}$  }
5. node\_y acknowledges  $K_{ij}$ ; compares  $K_{ji}$  with  $K_{ij}$ .  
node\_y : Check if  $K_{ij} = K_{ji}$

**Table 2.** Node-to-node mutual authentication

Sensor node_x		Sensor node_y
$L_{r_i}, U_{c_i}$	$\xrightarrow{U_{c_i}}$	$L_{r_j}, U_{c_j}$
$L_{r_i} \times U_{c_j} \cdot K_{ij}$	$\xleftarrow{U_{c_j}, K_{ji}}$	$L_{r_j} \times U_{c_i} \rightarrow K_{ji}$
$K_{ij} = K_{ji}$	$\xrightarrow{K_{ij}}$	$K_{ji} = K_{ij}$

### 4 Performance Analysis

A random graph  $G(n,p)$  is a graph of  $n$  nodes for which the probability that a link exists between two nodes is  $p$ . When  $p$  is zero, the graph does not have any edge, whereas when  $p$  is one, the graph is fully connected. Erdos and Renyi [12] showed that, for monotone properties, there exists a value of  $p$  such that the property moves from “nonexistent” to “certainly true” in a very large random graph. The function defining  $p$  is called the threshold function of the property. Given a desired probability  $P_c$  for graph connectivity, the threshold function  $p$  is defined by

$$P_c = \lim_{n \rightarrow \infty} P_r[G(n,p) \text{ is connected}] = e^{-e^{-pc}}, \text{ where } p = \frac{\ln(n) - \ln(-\ln(P_c))}{n} \tag{1}$$

Let  $p$  be the probability that a shared key exists between two sensor nodes,  $n$  be the number of nodes, and  $d$  be the expected degree as

$$d = p \times (n - 1) = \frac{(n - 1)(\ln(n) - \ln(-\ln(P_c)))}{n} \tag{2}$$

Figure 2 illustrates the plot of the expected degree of a node,  $d$ , as a function of the network size,  $n$ , for various values of  $P_c$ . The figure shows that the expected degree of a node needs to be increased by two to increase the probability that a random graph is connected by one order. Moreover, the curves of the plot are almost flat when  $n$  is large, indicating that size of the network has insignificant impact on the expected degree of a node required to have a connected graph.

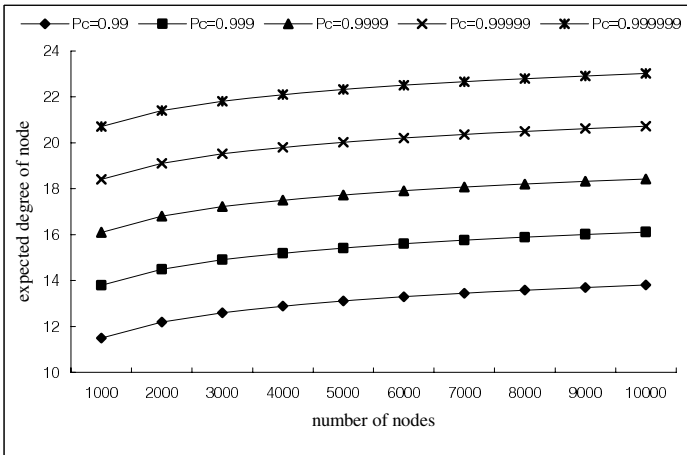


Fig. 2. Expected degree of a node for varying number of nodes

For a given density of sensor network deployment, let  $N$  be the expected number of neighbors within the communication range of a node. Using the expected node degree calculated above, the required local connectivity,  $P_{\text{required}}$ , can be estimated as follows [10].

$$P_{\text{required}} = \frac{d}{N} = \frac{(n-1)(\ln(n) - \ln(-\ln(P_c)))}{nN} \tag{3}$$

After we derive the required local connectivity, we decide the value  $S$  (the size of the key pool) and  $k$  (the number of keys in each node). The actual local connectivity is determined by these values. Note that  $S$  is not directly related to the sensor network, but  $k$  is related to the memory size of sensor node. Therefore,  $k$  needs to be as small as possible. We use  $P_{\text{actual}}$  to represent the actual local connectivity, which is the probability of any two neighboring nodes to find a common key between themselves. The link availability of any two nodes of the existing scheme [10] is then

$$1 - \text{Prob [a pair of nodes do not share a key]}. \tag{4}$$

The probability that a pair of nodes, A and B, do not share a common key can be found using  $P_{\text{actual}}$

$$P_{\text{actual}} = 1 - \frac{{}_S C_k \times {}_{S-k} C_k}{({}_S C_k)^2} = 1 - \frac{((S-k)!)^2}{S!(S-2k)!}. \tag{5}$$

Since,  $S$  is very large, we use Stirling's formula for  $n!$

$$n! \approx \sqrt{2\pi n} \left[ \frac{n}{e} \right]^n. \tag{6}$$

To simplify the expression of  $P_{\text{actual}}$ , it is approximated as follows.

$$P_{\text{actual}} = 1 - \frac{(P-k)^{2P-2k+1}}{(P-2k)^{P-2k+\frac{1}{2}}}. \tag{7}$$

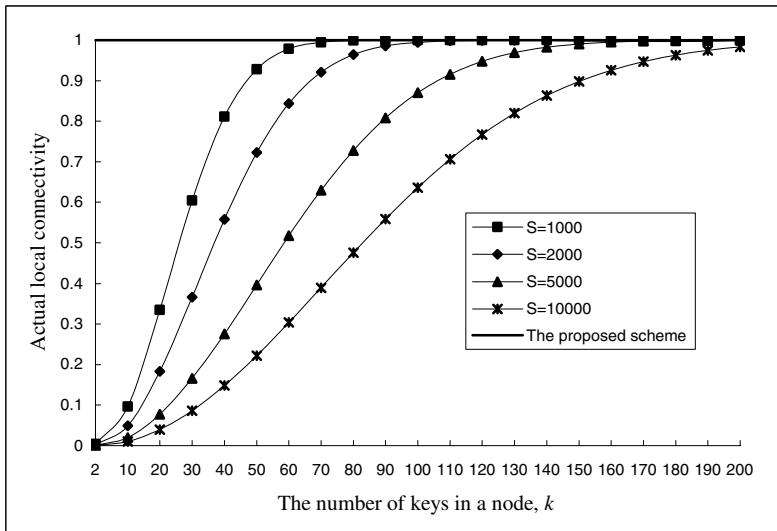


Fig. 3. Comparison of the connectivity of the proposed scheme with the existing scheme

Figure 3 compares the actual local connectivity of the proposed scheme with that of the existing scheme [10] when the size of the key varies from 2 to 200 for the size of the key pool  $S$  of 1000, 2000, 5000, and 10000. Observe from the figure that the local connectivity increases as the number of keys in a node increases for the existing scheme when the size of the pool of keys is fixed. The proposed scheme always allows the connectivity regardless of the number of keys per node. Note that, the superiority of the proposed scheme becomes more substantial when the memory size of the sensor node is small.

## 5 Conclusion and Future Works

Most earlier schemes proposed for security of distributed sensor network used asymmetric cryptography such as Diffie-Hellman key agreement or RSA. However, these schemes are often not suitable for distributed sensor network due to limited computation and energy resources of sensor node. The existing key pre-distribution scheme proposed to address this issue has a drawback of unguaranteed shared key between two nodes wanting to communicate. In this paper thus we have proposed a new key pre-distribution scheme guaranteeing that any pair of nodes can find a common key between themselves using the keys assigned by LU decomposition of a symmetric matrix. Also, it allows enhanced security by node-to-node mutual authentication. The existing scheme requires a large number of keys in each sensor node to display a comparable connectivity as the proposed scheme which allows 100% connectivity regardless of the number of keys. Therefore, the superiority of the proposed scheme is more substantial when the memory size of the sensor node is small. A new model considering not only connectivity but also security in a more formal way will be developed in the future.

## References

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci.: A survey on sensor networks: IEEE Communications Magazine, Vol. 40, no. 8, (2002) 102-114
- [2] David W. Carman, Peter S. Kruus, and Brian J. Matt.: Constraints and approaches for distributed sensor network security: NAI Labs Technical Report #00-010, (2000)
- [3] F. Stajano.: Security for Ubiquitous Computing: Jhon Wiley and Sons, ISBN 0-470-84493-0, (2002)
- [4] J. Rabaey, J. Ammer, J. L. da Silva, D. Patel.: PicoRadio, Ad hoc wireless networking of ubiquitous low-energy sensor/monitor nodes, Workshop on VLSI, (2000)
- [5] W. Du, J. Deng, Y. S. Han, S. Chen, and P. K. Varsheny.: A key management scheme for wireless sensor networks using deployment knowledge: Technical Report, Syracuse University, (2003)
- [6] R. Anderson and M. Kuhn.: Tamper resistance – a cautionary note: Proceeding of the Second Usenix Workshop On Electronic Commerce, (1996) 1-11
- [7] D. Liu and P. Ning.: Establishing pairwise keys in distributed sensor networks: Proceedings of the 10<sup>th</sup> ACM Conference on Computer and Communications Security, (2003) 52-61



- [8] C. Blundo, A. D. Santix, A. Herzberg, S. Kuten, U. Vaccaro, and M. Yung.: Perfectly-secure key distribution for dynamic conferences: Lecture Note in Computer Science, Vol. 740, (1993) 471-486
- [9] A. Perrig, R. Szewczyk, V. Wen, D. Cullar, and J. D. Tygar.: Spins (Security protocols for sensor networks): Proceeding of the 7<sup>th</sup> Annual ACM/IEEE International Conference on Mobile Computing and Networking(MobiCom), (2001) 189-199
- [10] L. Eschenauer and V. D. Gligor.: A key-management scheme for distributed sensor networks: Proceeding of the 9<sup>th</sup> ACM Conference on Computer and Communication security, (2002) 41-47
- [11] H. Chan, A. Perrig, and D. Song.: Random key pre-distribution schemes for sensor networks: IEEE Symposium on Security and Privacy, (2003) 197-213
- [12] Erdos and Renyi.: On random graphs I: Publ. Math. Debrecen, Vol 6, (1959) 290-297
- [13] Sung Jin Choi, Hee Yong Youn.: A Novel Data Encryption and Distribution Approach for High Security and Availability Using LU Decomposition: The 2004 International Conference On Computational Science And Its Applications, LNCS 3046, (2004), 637-646

# Energy-Driven Adaptive Clustering Hierarchy (EDACH) for Wireless Sensor Networks\*

Kyung Tae Kim and Hee Yong Youn\*\*

School of Information and Communications Engineering,  
Sungkyunkwan University, Suwon, Korea  
harisu@skku.edu, youn@ece.skku.ac.kr

**Abstract.** Wireless sensor network consists of small battery powered sensors. Therefore, energy consumption is an important issue and several schemes have been proposed to improve the lifetime of the network. In this paper we propose a new approach called energy-driven adaptive clustering hierarchy (EDACH), which evenly distributes the energy dissipation among the sensor nodes to maximize the network lifetime. This is achieved by using proxy node replacing the cluster-head of low battery power and forming more clusters in the region relatively far from the base station. Comparison with the existing schemes such as LEACH (Low-Energy Adaptive Clustering Hierarchy) and PEACH (Proxy-Enabled Adaptive Clustering Hierarchy) reveals that the proposed EDACH approach significantly improves the network lifetime.

**Keywords:** Cluster-head, energy consumption, network lifetime, proxy node, wireless sensor networks.

## 1 Introduction

Wireless sensor networks have been evolving rapidly and now they are widely used in both the military and civilian applications such as target tracking, surveillance, and security management [1,2]. It is composed by hundreds or thousands of low-cost, low power, multifunctional small sensor nodes. Since a sensor is a small, lightweight, untethered, battery-powered device, it has limited energy. Therefore, energy consumption is a critical issue in wireless sensor networks [4,5]. In a sensor network a large number of sensors are deployed to carry out a given task. When sensors are deployed in the field, they establish routes and then start sensing the surroundings, computation and transmission of the data to the base station. Since sensor nodes carry limited, generally irreplaceable power source, the sensor network protocols must focus primarily on power conservation [6,7,10].

The protocol called low-energy adaptive clustering hierarchy (LEACH) [3] is a cluster-based protocol proposed to solve the energy consumption problem. It equally distributes the entire energy consumption of a sensor network among the sensors, and

---

\* This research was supported by the Ubiquitous Autonomic Computing and Network Project, 21st Century Frontier R&D Program in Korea and the Brain Korea 21 Project in 2005.

\*\*Corresponding author.

thus increases the lifetime of the network. In LEACH, however, a cluster-head can cause a failure because of energy deficiency. Sensors in a sensor network are susceptible to failures due to limited battery power and will also be inactive regardless of their power condition if the cluster-head in their cluster suffers from the energy deficiency failure. If a failure occurs at the cluster-head, the network has to be re-clustered and a new schedule is transmitted to the sensors. Also, the sensor network has to stop data processing and communication in order to perform re-clustering, which requires network setup and bootstrapping. Consequently, frequent failures will result in large re-clustering overhead of energy and time. Lifetime of a sensor network and its energy efficiency are thus greatly affected by the frequency of re-clustering.

PEACH (Proxy-Enabled Adaptive Clustering Hierarchy) [11] is a protocol that improves LEACH in terms of network lifetime. This is achieved by selecting a proxy node which can assume the role of the current cluster-head of weak power during one round of communication. PEACH is based on the consensus of healthy nodes for the detection and manipulation of failure in any cluster-head. It allows considerable improvement in the network lifetime by reducing the overhead of re-clustering.

In LEACH and PEACH, however, the cluster-heads are selected randomly with uniform distribution in the network. Note that energy consumption of a cluster-head will be much larger than that of cluster member nodes for the communication with the nodes inside its cluster and, more importantly, the base station. This is because energy consumption is proportional to the distance between the communicating nodes, while the base station is usually much farther than the cluster member nodes from a cluster-head. If the cluster-heads are uniformly laid out in the entire region of the sensor network, then the ones relatively far from the base station than the other cluster-heads will suffer from the energy deficiency problem more likely. Therefore, in this paper, we propose the energy-driven adaptive clustering hierarchy (EDACH) approach that puts more number of cluster-heads in the region relatively far from the base station. The number of member nodes in their clusters will then be smaller than that of other clusters. This compensates the larger energy consumption due to larger distance to the base station. It still employs the same proxy node approach as in PEACH to solve the problem of cluster-head having insufficient energy for carrying out the duty of cluster-head. If a cluster encounters a problematic cluster-head, then a proxy is selected to operate in replace of the original cluster-head. Computer simulation reveals that the proposed EDACH approach extends the network lifetime of LEACH and PEACH about 80% and 30%, respectively.

The remainder of the paper is organized as follows. In Section 2 the system model including the energy model and fault model is presented along with the related works. Section 3 presents the proposed EDACH scheme. Section 4 evaluates the performance of the proposed scheme by computer simulation, and compares it with LEACH and PEACH. Finally, Section 5 concludes the paper and outlines future research directions.

## 2 The System Model

In this paper the single-hop sensor network architecture is employed for wireless sensor network as LEACH [3]. There exist three kinds of nodes in the network; sensor nodes, cluster-head nodes, and base station. The sensor nodes are used for data acquisition. The cluster-heads are for data fusion and forwarding of the aggregated infor-

mation toward the base station. As a result, the cluster-heads consume energy at a substantially higher rate than the other nodes due to wireless communication over large distances. Consequently, the cluster-heads have shorter lifetime. Upon depletion of energy at a cluster-head, the coverage for the particular area under surveillance by that node is lost. The base station may be assumed to always have sufficient battery provisioning, or its battery may be re-provided during its course of operation. Therefore, its power consumption is not a concern in our investigation.

## 2.1 The Energy Model of a Sensor

We use the same radio model as discussed in [3, 8, 11], which is the first order radio model. In this model, a radio dissipates  $E_{elec}$  ( $=50$  nJ/bit) to run the transmitter or receiver circuitry and  $\xi_{mp}$  ( $=100$  pJ/bit/m<sup>2</sup>) for the transmitter amplifier. The energy consumption model is described as follows. For transmission, when a node transmits  $k$ -bit data to another node with a distance of  $d$ , the energy it consumes is  $E_{Tx}(k, d) = E_{elec} \times k + \xi_{mp} \times k \times d^2$ . For receiving, when a node receives  $k$ -bit data, the energy it consumes is  $E_{Rx}(k) = E_{elec} \times k$ . Here  $E_{elec}$  is the energy required for transmitting or receiving one bit data, the second term of  $E_{Tx}$  covers the energy loss due to channel attenuation, and  $\xi_{mp}$  is the amplifier coefficient. For simplicity of calculation, we assume that transmission range of each node is same on one condition that the range should cover all the neighbors in its cluster. We also assume that all data packets are same sizes. For fair comparison, we use the same constant coefficients as in LEACH.

## 2.2 The Fault Model

Data transmission failure occurs when a cluster-head cannot transmit data due to energy deficiency. Failure at a cluster-head affects the system status and causes a remedial operation such as re-clustering or boot-strapping that results in increased network down time and reduced network lifetime, etc. We assume that the data in the communication are error-free and the semantic-related generic faults during data transmission can be detected and removed by the application specific operation. Data transmission failure at a cluster-head can be caused by hardware failure or energy deficiency.

Failure at a cluster-head prevents it from transmitting data to the sensors as well as relaying the data to the base station. The data sent by the sensor nodes will also be lost. Once a cluster-head fails, it can no longer serve as a liaison between the sensor nodes and the base station. The failure at the cluster-head is tried to be avoided using the proxy nodes and non-uniform distribution of the cluster-heads as explained later. In our fault model we consider only the failure at the cluster-heads.

With LEACH, the cluster-heads are stochastically selected. In order to select the cluster-heads, each node generates a random number between 0 and 1. If the number is smaller than the threshold,  $T$ , the node becomes a cluster-head for the current round. The threshold is calculated as follows.

$$T = \frac{P}{1 - P \times (r \bmod \frac{1}{P})} \quad (1)$$

Here  $P$  is the portion of the nodes becoming the cluster-heads and  $r$  is the number of current round. If a node once has been a cluster-head in the last  $1/P$  rounds, it cannot be a cluster-head again. This algorithm thus ensures that every node becomes a cluster-head exactly once within  $1/P$  rounds.

Refer to Figure 1(a) of an example of clustering with LEACH and PEACH, where five cluster-heads (the black dots) are selected randomly among 100 nodes with uniform distribution. Here the base station is assumed to be located outside the region at the upper-right corner direction. Since the cluster-heads were selected randomly with uniform distribution, the ones far from the base station will consume their energy more quickly than the others as mentioned earlier. Therefore, we propose to put more cluster-heads in the area of the nodes far from the base station. This can be easily implemented by setting the  $P$  value of Equation (1) differently. For example, as shown in Figure 1(b), the entire range is partitioned into three segments, and the  $P$  value of the nodes in the segment close or far from the base station is decremented or incremented, respectively. This approach is expected to evenly distribute energy consumption among the nodes, and thus outperform the existing approaches having uniform distribution of cluster-heads. The proposed scheme is presented next.

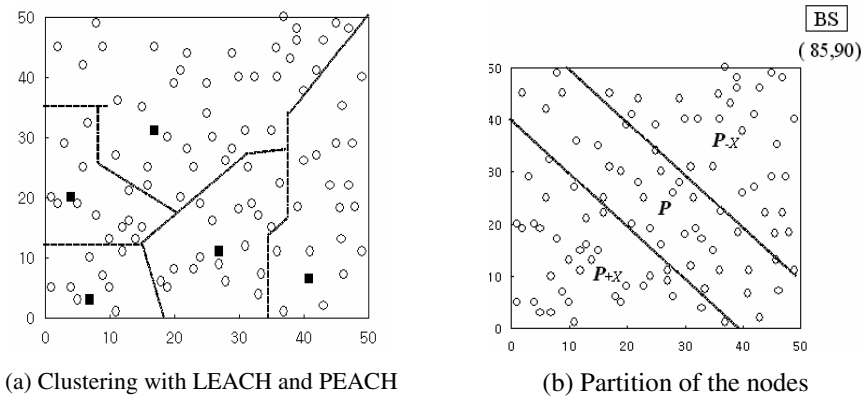


Fig. 1. An example of clustering and node partitioning

### 3 The Proposed Scheme

We call the proposed protocol as energy-driven adaptive clustering hierarchy (EDACH), which is an enhanced version of the LEACH and PEACH scheme. One round of its operation consists of the following two phases: i) set-up phase, ii) self-organized data collection and transmission phase. The proposed scheme is to solve the possible problem of the cluster-head in the LEACH approach having insufficient energy for carrying out the duty of cluster-head by using proxy nodes like the PEACH approach. It, however, further improves the performance of PEACH by forming more clusters in the region relatively far from the base station.

Each round of communication in the proposed scheme begins with the set-up phase where the clusters are organized, followed by the self-organized data collection and

transmission phase where data transfer to the base station occurs. The second phase also includes the proxy node selection process and Indicator Control Message (ICM) advertisement process. We next introduce the EDACH protocol in detail.

### 3.1 The Set-Up Phase

In LEACH and PEACH, a fixed portion of the sensors stochastically decides themselves as cluster-heads as described in [3,11]. The problem of cluster-head population in a wireless sensor network was analyzed in [3]. They showed that 5% of the nodes in the network operating as cluster-heads can achieve good performance in a homogeneous network with various parameter settings. Recall that we propose to regulate the number of cluster-heads according to the proximity to the base station. For this, we partition the entire area of the sensor network into three segments as near, medium, and far segment as shown in Figure 1(b). Of course, the shape of partition will be different according to the position of the base station, and the number of segments could also be varied.

In the set-up phase, each node calculates the threshold using Equation (1), but the values of  $P$  for the nodes belonging to different segments differ from each other. That is, the  $P$  value for the nodes in the near, medium, and far segment is  $(1-x)P$ ,  $P$ , and  $(1+x)P$ , respectively, where  $0 < x < 1$ . With this arrangement the nodes relatively close to the base station will have smaller threshold value and thus smaller chance to become a cluster-head. As a result, the number of cluster-heads in that segment will be relatively small. Computer simulation reveals that the proposed approach increases the network lifetime compared to PEACH. Figure 2 shows the result of clustering with EDACH for the same distribution of nodes as in Figure 1.

When a node is selected as a cluster-head, it generates a cluster-head token. Then, the selected cluster-head advertises its token by CSMA/CA MAC protocol to all its neighbors. After the remaining nodes receive the advertisements, they compare the strengths of the received signals. It keeps only the token with the strongest signal in every comparison, and randomly chooses a one if a tie occurs.

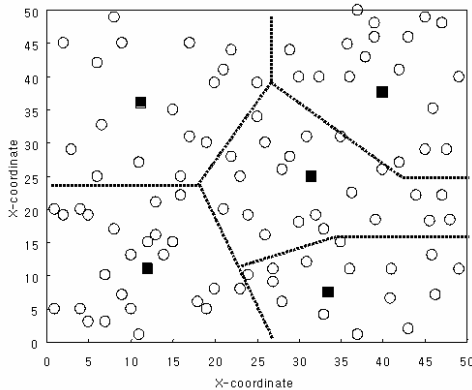


Fig. 2. An example of clustering with EDACH after the set-up phase

After the advertisement, every cluster member node recognizes the source of the token as its cluster-head and broadcasts the topology reply packet by CSMA/CA MAC protocol back to the cluster-head. In the reply packet the node's position (NP) and remaining energy (RE) level are included. When the cluster-heads receive the reply packets, they set up a schedule for the nodes in their cluster. Based on the number of nodes in the cluster, the cluster-head creates a TDMA schedule indicating when each node in the cluster can transmit.

### 3.2 The Self-organized Data Collection and Transmission Phase

After the set-up phase, the self-organized data collection and transmission phase starts. Every sensor node collects the data and then sends a packet to the cluster-head in its scheduled transmission time. The radio of a node is turned off when it is outside its scheduled time slot in order to save the energy. Each cluster-head keeps its receiver on to collect data from the nodes in its cluster and continuously updates the table listing the energy of the nodes based on the received packets. When the data from all the nodes have been received, the cluster-head executes the function for data fusion to aggregate the received data into one packet. After the data fusion, the cluster-head sends it to the base station. As a cluster-head needs to receive many packets and consume large power for long range transmission, its energy is used up more quickly than other nodes in the cluster. Therefore, a cluster-head can cause a failure because of energy deficiency. If a failure occurs at a cluster-head, the network has to be re-clustered and a new schedule needs to be transmitted to the sensors. This will significantly reduce the network lifetime. In order to solve this problem, a proxy node is selected if the battery power of the cluster-head becomes smaller than the threshold explained next.

**Calculation of the Threshold Value.** The threshold value,  $E_{TH}$ , plays a very important role in the data transmission phase since it is used as a measure for deciding if the current cluster-head has become obsolete. It is calculated when a cluster-head is selected. If the energy of the cluster-head drops below the threshold, the proxy node selection process begins. We assume that all sensors are identical and produce data at the same rate. The following equations are used to compute the threshold value.

$$k_j = M_{bit} \sum_{i=1}^{\lfloor \frac{n}{k} \rfloor - 1} k_{ij} \quad (2)$$

$$E_{CH(j)} = E_{elec} \times k_j + \varepsilon_{amp} \times k_j \times d_{CH(j)}^2 \quad (3)$$

$$E_{TH} = \frac{1}{k} \sum_{j=1}^k E_{CH(j)} \quad (4)$$

Here  $k_j$  is the length of the aggregated message in the cluster-head and  $d_{CH}$  is the distance between the cluster-head and base station. Since  $E_{TH}$  changes with time, it is recalculated in every self-organized data collection and transmission phase.

When the energy level of the cluster-head falls below the threshold, a proxy node is selected using the RE and NP value of the reply packets received in the set-up phase (Refer to [11]). After a node is selected as a proxy node, the cluster-head

broadcasts an indicator control message (ICM) containing the address of the proxy node and a new TDMA schedule for the member nodes. The member nodes receiving the ICM send a confirmation message to the proxy node. When the proxy node receives the confirmation message, it keeps the node IDs of the member nodes. After the message exchanges are over, the member nodes resume data transmission. The ICM advertisement process is described in Figure 3.

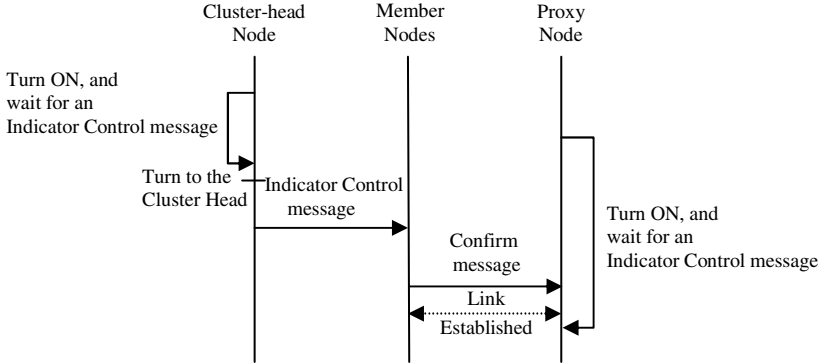


Fig. 3. The sequence of operations in the ICM advertisement process

### 4 Performance Evaluation

We evaluate the effectiveness of the proposed EDACH scheme along with LEACH and PEACH through computer simulation. The probability for a node to be selected as a cluster-head is set to 5%. For the simulation we consider a sensor network of 100 sensor nodes randomly arranged in a  $50 \times 50$  region. A base station is located at (85, 90). We use two models of initial residual energy of sensor nodes; uniform at 0.5J and random between 0.25J and 0.5J. We set  $E_{elec}$  to 50 (nJ/bit) and  $\epsilon_{amp}$  to 100 (pJ/bit/m<sup>2</sup>) in the energy model of a sensor. The size of sensor data is 2000 bits, and the advertisement message is 64-bit long. The value of  $x$  in the  $(1-x)P$  and  $(1+x)P$  formula mentioned earlier is set to 0.2. In the simulation the result of 100,000 runs are averaged.

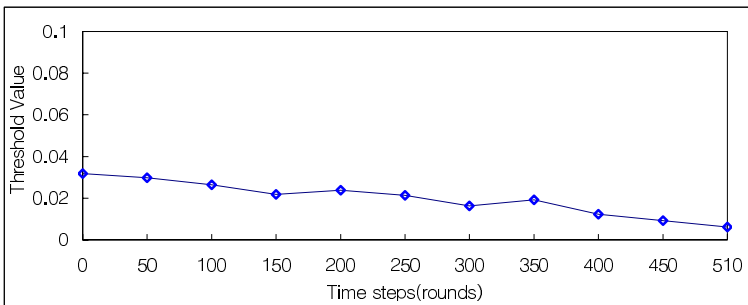


Fig. 4. The threshold values obtained as time moves



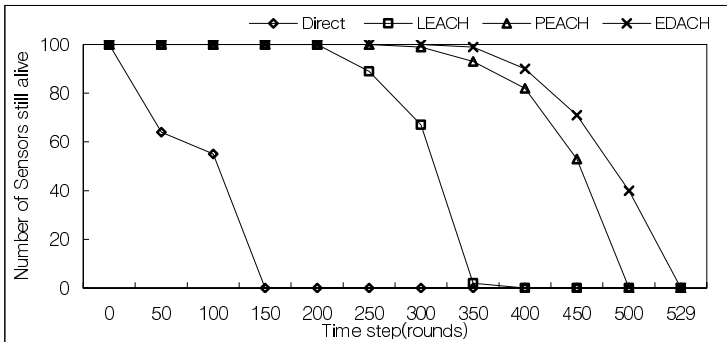
Figure 4 shows the threshold values obtained as time moves. When  $E_{TH}$  is large, a proxy node is more likely selected to take over the role of the cluster-head and thus the cluster-head can reserve energy for later use. In this sense proxy node can help the cluster-head have a longer life.

Table 1 lists the lifetime of the sensor network in terms of the round a node begins to die and the round the last node dies for the three schemes compared. Notice that the proposed EDACH protocol is consistently better than the others. Especially, the proposed EDACH outperforms LEACH and PEACH more significantly when the initial energy is relatively high. We ran the simulator with different energy thresholds and obtained similar results.

**Table 1.** The network lifetimes with different initial energies of the sensors

Energy (J/node)	Protocol	The round a node begins to die	The round a node begins to dies
0.25	Direct	52	112
	LEACH	228	353
	PEACH	294	492
	EDACH	347	529
0.5	Direct	104	211
	LEACH	498	756
	PEACH	795	1029
	EDACH	953	1358

The improvement offered by the proposed EDACH protocol over LEACH and PEACH can be clearly seen in Figure 5, which shows the number of sensors alive as the round proceeds with 0.25 J/node initially. A sensor node with insufficient residual energy can occasionally become a cluster-head even though there is a sensor node with more battery power nearby. It then exhausts the energy, stops operating, and disrupts gathering data in its cluster. Also, data transmission to the base station is not possible. On the other hand, in the proposed protocol, the approach of proxy node and distribution of cluster-heads considering the distance to the base station allows significantly increased network lifetime.



**Fig. 5.** Comparison of the number of live sensors as the round proceeds

Another important aspect of the proposed protocol is illustrated in Figure 6, which shows the locations of live (circle) and dead (dot) sensor nodes with PEACH and EDACH, respectively, after 480 rounds. Observe that, in addition to a lot more live nodes than PEACH, EDACH displays well dispersed live nodes. This will be an important aspect that the proposed EDACH protocol can avoid dead spot and extend the lifetime of the network. This was achieved by employing the proxy node approach for weak cluster-heads and non-uniform distribution of them.

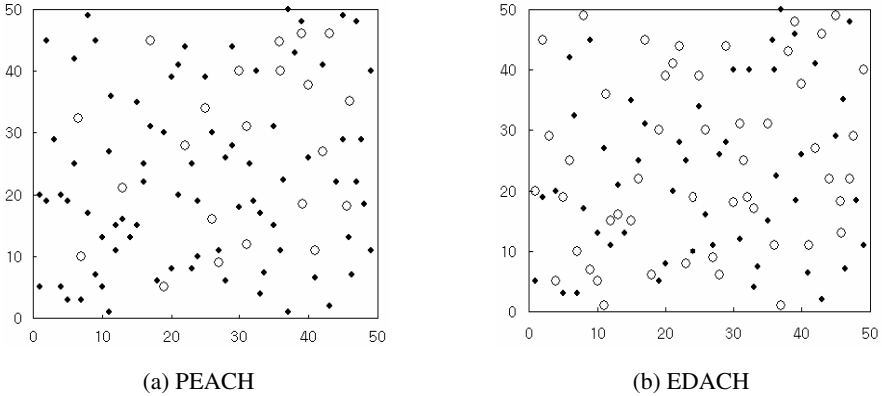


Fig. 6. The distribution of live (circle) and dead (dot) nodes after 480 rounds

### 5 Conclusion and Future Work

In this paper we have proposed a protocol called EDACH that solves the problem of cluster-head having low energy using the proxy node approach which assumes the role of the cluster-head in the wireless sensor network. The proposed protocol further enhances the network lifetime by distributing the cluster-heads according to the distance to the base station. Computer simulation results showed that the proposed approach allows much longer lifetime of wireless sensor network than the existing schemes. The proposed approach will be more important when the wireless sensor network is deployed in large area and the base station is far from the network.

The future work will focus on the comparison of the EDACH approach with other approaches such as simulated annealing and taboos search. In the proposed EDACH approach several factors have been decided heuristically such as the number of partitioned segments, the threshold energy value used for deciding whether a proxy node is required or not, etc. A formal methodology will be developed in order to determine such factors in a more systematic way and also to allow optimal values for the given conditions. In addition, the proposed approach will be extended to multi-level cluster hierarchy.

## References

- [1] L. Zhong, R. Shah, C. Guo, J. Rabaey: An ultra low power and distributed access protocol for broadband wireless sensor networks: IEEE Broadband Wireless Summit, Las Vegas, May 2001.
- [2] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci.: Wireless sensor networks: a survey *Computer Networks*: 38 (4) (2002) pp. 393-422
- [3] W.R.Heinzelman, A.Chandrakasan, and H. Balakrishnan.: Energy-Efficient Communication Protocol for Wireless Micro-sensor Networks: In Proceedings of the Hawaii International Conference on System Science, Maui, Hawaii,2000.
- [4] A. Sinha, A. Chandrakasan.: Dynamic power management in wireless sensor networks: IEEE Design & Test of Computers, March-April 2001, S. 62-74.
- [5] L. Zhong, R. Shah, C. Guo, J. Rabaey.: An ultra low power and distributed access protocol for broadband wireless sensor networks: IEEE Broadband Wireless Summit, May 2001.
- [6] G. Gupta and M. Younis.: Fault-Tolerant Clustering of Wireless Sensor Networks: IEEE Wireless Communications and Networking, Vol. 3 pp. 1577-1584, Mar 2003.
- [7] K. Sohrabi, J. Gao, V. Ailawadhi, G.J. Pottie.: Protocols for selforganization of a wireless sensor Network: IEEE Personal Communications, October 2000, pp. 16–27.
- [8] W. Heinzelman, A. Chandrakasan, H. Balakrishnan.: An applicationspecific protocol architecture for wireless microsensor networks: in press: IEEE Trans. on Wireless Networking.
- [9] G. Gupta and M. Younis.: Load-balanced clustering of wireless sensor networks: IEEE ICC, 1848-1852, May 2003.
- [10] MJ Handy, M. Haase, D. Timmermann.: Low-Energy Adaptive Clustering Hierarchy with Deterministic Cluster-Head Selection: August 2002.
- [11] K.T. Kim and H.Y. Youn.: PEACH: Proxy-Enable Adaptive Clustering Hierarchy for Wireless Sensor network: Proceeding of The 2005 International Conference On Wireless Network, June 2005, pp. 52-57.

# A Load-Balancing and Energy-Aware Clustering Algorithm in Wireless Ad-Hoc Networks

Wang Jin, Shu Lei, Jinsung Cho, Young-Koo Lee,  
Sungyoung Lee\*, and Yonil Zhong

Department of Computer Engineering, Kyung Hee University, Korea  
{wangjin, sl8132, sylee, zhungs}@oslab.khu.ac.kr  
{chojs, yklee}@khu.ac.kr

**Abstract.** Wireless ad-hoc network is a collection of wireless mobile nodes dynamically forming a temporary communication network without the use of any existing infrastructure or centralized administration. It is characterized by both highly dynamic network topology and limited energy. So, the efficiency of MANET depends not only on its control protocol, but also on its topology and energy management. Clustering strategy can improve the performance of flexibility and scalability in the network. With the aid of graph theory, genetic algorithm and simulated annealing hybrid optimization algorithm, this paper proposes a new clustering strategy to perform topology management and energy conservation. Performance comparison is made between the original algorithms and our two new algorithms, namely an improved weighting clustering algorithm and a novel Genetic Annealing based Clustering Algorithm (GACA), in the aspects of average cluster number, topology stability, load-balancing and network lifetime. The experimental results show that our clustering algorithms have a better performance on average.

## 1 Introduction

Wireless ad hoc wireless network is a collection of wireless mobile nodes that self-configure to form a network without the aid of any established infrastructure [1]. It can be rapidly deployed and reconfigured where the communication infrastructure is either unavailable or destroyed. However, it is confronted with many challenges too, such as the mobility of hosts, the dynamic topology, the multi-hop nature in transmission, the limited bandwidth and battery, etc. So, the study of MANET (Mobile Ad-hoc NETWORK) is a very demanding and challenging task.

Up to now, there are many routing protocols based on various strategies in MANET, and they can be classified into several kinds as follows: (1) proactive and reactive; (2) flat and hierarchical; (3) GPS assisted and non-GPS assisted, etc. These kinds of protocols can be used solely or together. Here we mainly discuss the hierarchical routing protocols, which are based on the clustering algorithm [2, 3].

The rest of the paper is organized as follows. In section 2, some relevant background and commonly used clustering algorithms are presented. Based on which, an

---

\* Corresponding author.

improved clustering algorithm is proposed in section 3. In section 4, another novel Genetic Annealing based Clustering Algorithm (GACA) is given so as to optimize the overall network performance. The simulation results and comparison is made in the aspects of average cluster number, topology stability, load-balancing and network lifetime in section 5. Section 6 concludes the paper.

## 2 Related Work

Similar to the cellular network, the MANET can be divided into several clusters. Each cluster is composed of one clusterhead and many normal nodes, and all the clusterheads form an entire dominant set. The clusterhead is in charge of collecting information (signaling, message, etc.) and allocating resources within its cluster and communicating with other clusterheads. And the normal nodes communicate with each other through their clusterhead, no matter they are in the same cluster or not.

Several original clustering algorithms have been proposed in MANET. These include: (1) Highest-Degree Algorithm; (2) Lowest-ID Algorithm; (3) Node-weight Algorithm; (4) Weighted Clustering Algorithm. (5) Others, like RCC (Random Competition based Clustering), LCC (Least Cluster Change), LEACH etc. We will give some of them a brief description as follows.

### 2.1 Highest-Degree Algorithm

The Highest-Degree Algorithm was originally proposed by Gerla and Parekh [4,5]. A node  $x$  is considered to be a neighbor of another node  $y$  if  $x$  lies within the transmission range of  $y$ . The node with maximum number of neighbors (i.e., maximum degree) is chosen as a clusterhead.

Experiments demonstrate that the system has a low rate of clusterhead change but the throughput is low under the Highest-Degree Algorithm. As the number of nodes in a cluster increases, the throughput drops and hence a gradual degradation in the system performance is caused. All these drawbacks occur because this approach does not have any restriction on the upper bound of node degree in a cluster.

### 2.2 Lowest-ID Algorithm

This Lowest-ID Algorithm was originally proposed by Baker and Ephremides [6]. It assigns a unique id to each node and chooses the node with the minimum id as a clusterhead.

As for this algorithm, the system performance is better compared with the Highest-Degree Algorithm in terms of throughput. But it does not attempt to balance the load uniformly across all the nodes.

### 2.3 Node-Weight Algorithm

Basagni et al. [7] proposed two algorithms, namely distributed clustering algorithm (DCA) and distributed mobility adaptive clustering algorithm (DMAC). In these two approaches, each node is assigned a weight based on its suitability of being a clusterhead. A node is chosen to be a clusterhead if its weight is higher than any of its neighbor's weight; otherwise, it joins a neighboring clusterhead.

Results show that the number of updates required is smaller than the Highest-Degree and Lowest-ID Algorithms. Since node weights vary in each simulation cycle, computing the clusterheads becomes very expensive and there are no optimizations on the system parameters such as throughput and power control.

## 2.4 Weighted Clustering Algorithm

The Weighted Clustering Algorithm (WCA) was originally proposed by M. Chatterjee et al.[8]. It takes four factors into consideration and makes the selection of clusterhead and maintenance of cluster more reasonable. As is shown in equation (1), the four factors are node degree difference, distance summation to all its neighboring nodes, velocity and remaining battery power respectively. And their corresponding weights are  $w_1$  to  $w_4$ . Besides, it converts the clustering problem into an optimization problem and an objective function is formulated.

$$W_i = w_1\Delta_i + w_2D_i + w_3V_i + w_4E_i \quad (1)$$

However, only those nodes whose neighbor number is less than a fixed threshold value can be selected as a clusterhead in WCA. It is not very desirable in the practical application. For example, many well-connected nodes whose neighbor number is larger than the fixed threshold might be a good candidate as well. Besides, its energy model is too simple. It treats the clusterhead and the normal nodes equally and its remaining power is a linear function of time, which is also not very desirable. So, we proposed an improved clustering algorithm as follows.

## 3 The Improved Weighted Clustering Algorithm

From the discussion mentioned above, we can see that most clustering algorithms, except for the WCA, only take one of the following factors into consideration, such as the node degree, ID, speed or remaining power. When the problem in one aspect is solved, some other problems are introduced simultaneously. Inspired by the basic idea of WCA, we proposed an improved clustering algorithm.

On the one hand, WCA only chooses those nodes whose neighbor number is less than a fixed threshold as a clusterhead candidate. However, many well-connected nodes whose neighbor number is larger than the fixed threshold might be a good candidate as well. So we can also treat them as clusterheads candidates and select an affordable number of normal nodes from their neighboring nodes. On the other hand, we established a more practical energy-consumption model which we will explain later.

By solving the optimization problem of  $\min(W_i)$ , the clusterheads and their affiliated normal nodes are selected and a trade-off is made from four aspects.

### 3.1 Principles of the Improved Weighted Clustering Algorithm

In order to determine the fitness value  $W_i$  of a node as a clusterhead, we need to consider from the following four aspects.

If the node degree is higher, then the node is more stable as a clusterhead. Here we make a simple conversion  $\Delta_i = |N_i - M|$ , where  $N_i$  is the practical degree of node  $i$  and  $M$  is the maximum degree. The smaller  $\Delta_i$  is, the better node  $i$  will be as a clusterhead. As for those nodes whose practical degree is larger than the maximum degree  $M$ , we also treat them as clusterhead candidates. Once they are chosen as clusterheads, we will choose  $M$  nodes with less  $W_i$  as their normal nodes. It is a distinctive difference between the original WCA and our improved algorithm, and it can work very well under densely deployed ad hoc networks where the WCA becomes useless.

If the node velocity  $V_i$  is lower, then the node will be more stable as a clusterhead.

If the distance summation of node  $i$  to all its neighbors  $D_i$  is smaller, it will consume less transmission power to communication with the normal nodes within its cluster. In other words, the cost will be smaller.

If the remaining battery power  $E_i$  is higher, the longer it will be for node  $i$  to serve as a clusterhead. Here we make another conversion and set an energy-consuming model. All the  $E_i$ s are set to zero initiatorily. If the node serves as a clusterhead, we assume that it consumes 0.1 unit of energy and if normal node, 0.02 unit of energy. Once some  $E_i$  is above 1 (normalized), we believe that this node is out of energy and the network will become useless rapidly due to the avalanche effect [9]. The energy-consuming relationship of 5:1 is commonly used among some papers. And it meets with the minimization problem very well. As for some specific application, one can infer to the related technical report, such as the Mica2 Motes [10]. And the model is also applicable through minor modification.

### 3.2 Steps of the Proposed Algorithm

Taking node  $i$  as an example, we compute its  $W_i$  according to the following steps and then judge whether it is a clusterhead or a normal node.

Step 1: Compute its practical degree and then derive the equation  $\Delta_i = |N_i - M|$ .

Step 2: Compute the distance summation  $D_i$  to its neighboring nodes.

Step 3: Set the velocity  $V_i$  according to the random waypoint mobility model.

Step 4: At first, set  $E_i$  to zero and increase their values according to the energy-consuming model. Our algorithm terminates once some  $E_i$  is above 1 (normalized).

Step 5: Compute  $W_i$  according to various  $w_i$  under different application.

Step 6: Taking the node with minimum  $W_i$  as the first clusterhead and its neighboring nodes as its normal nodes within the same cluster. Then we go on with this process until all nodes act as either clusterheads or normal nodes.

Step 7: All the nodes move randomly after some unit time and it goes back to step 1 again. And it terminates until a maximum number of time is reached or some node is out of energy.

#### 4 A Novel Genetic Annealing Based Clustering Algorithm (GACA)

The selection of clusterheads set, which is also called dominant set in Graphic Theory, is a NP-hard problem. Therefore, it is very difficult to find a global optimum. So, we can take a further step to use the computational intelligence methods, such as Genetic Algorithm (GA) or Simulated Annealing (SA), to optimize the objective function.

Considering the length of our paper, we will skip the principles of GA and SA, and explain the steps of our new Genetic Annealing based Clustering Algorithm (GACA) directly.

The steps of our GACA are as follows. And it usually takes 5 to 10 iterations to convergence. So, we can say that it converges very fast.

Step 1: As for N nodes, randomly generate L integer arrangements in the range of [1, N].

Step 2: According to these random arrangements and the clustering principle of WCA, derive L sets of clusterheads and compute their corresponding  $\sum w_{iold}$ .

Step 3: According to the Roulette Wheel Selection and Elitism in GA, select L sets of clusterheads which are better, and replace the original ones.

Step 4: As for each of the L sets of clusterheads, perform the crossover operator and derive the new L sets of clusterheads and their  $\sum w_{inew}$ .

Step 5: According to the Metropolis “accept or reject” criteria in SA, decide whether to take the one from L sets of clusterheads in  $\sum w_{iold}$  or in  $\sum w_{inew}$ . And the new L sets of clusterheads in the next generation are obtained.

Step 6: Repeat Step 3 to 5 until it converges or a certain number of iteration is reached. And in our simulation, it usually takes 5 to 10 iterations to converge.

Then the global optimal or sub-optimal solution  $\min (\sum w_{inew}) (i=1, 2 \dots L)$  is obtained and their corresponding set of clusterheads is known.

In Step 2, we make L random arrangements in order to reduce the randomness in the clustering process, because there is much difference in the set of clusterheads (or dominant set) for different nodes arrangements.

As for the Roulette Wheel Selection in Step 3, we do not take the traditional selection probability  $P_i = \frac{\sum w_i}{\sum_{i=1}^L (\sum w_i)}$ , but  $P_i = \frac{e^{-\sum w_i}}{\sum_{i=1}^L (e^{-\sum w_i})}$ . In that case, the set of



clusterheads whose  $\sum w_i$  is smaller will have more chance to be selected. Besides, to overcome the randomness in the probability problem, we preserve the best set of  $\sum w_{iold}$  directly to the  $\sum w_{inew}$  in Elitism.

To further reduce the randomness and increase the probability that the global solution may occur, we perform M pairs of crossovers as for each of L random arranged integers (i.e. mobile nodes). And the new L sets of clusterheads and their  $\sum w_{inew}$  (i=1,2,...L) are derived.

In Step 5, we make the “accept or reject” decision according to the Metropolis criteria. If  $\sum w_{inew} \leq \sum w_{iold}$ , then we accept  $\sum w_{inew}$  directly. If  $\sum w_{inew} > \sum w_{iold}$ , we do not reject it directly, but accept it with some probability.

In other words, if  $e^{-\frac{\sum w_{inew} - \sum w_{iold}}{\alpha T}}$  is larger than a randomly generated number in the range of (0,1), which shows that  $\sum w_{inew}$  and  $\sum w_{iold}$  may be very close to each other, we will still take it. Or else, we will reject the one in  $\sum w_{inew}$  and take its counterpart in  $\sum w_{iold}$ . Besides, we let  $T = \alpha T$  ( $\alpha$  is a constant between 0 and 1 and we normally take 0.9) after each iteration, so that  $\sum w_{inew}$  and  $\sum w_{iold}$  must be closer if  $\sum w_{inew}$  is to be accepted. In this way, our GACA will not be trapped in the local optima and the premature effect can be avoided. In other words, the diversity of searching space can be ensured and it is similar to the mutation operator in GA.

## 5 Performance Evaluation

We set our simulation environment as follows. There are N nodes randomly placed within a range of 100 by 100 m<sup>2</sup>, whose transmission range varies from 15m to 50m. A Random Waypoint mobility model is adopted here. And our GACA parameters are listed in table 1.

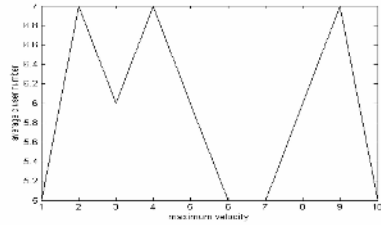
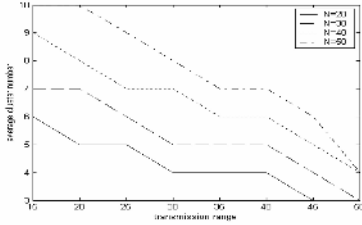
**Table 1.** GACA parameters

M	L	$\alpha$	$\mathcal{E}$
1	10	0.9	0.01

### 5.1 Analysis of Average Cluster Number

As is shown in figure 1, we simulate N nodes whose transmission range varies from 15m to 50m. We can conclude that:

- (1) The average cluster number (ACN) decreases as the transmission range increases.
- (2) As for a smaller transmission range, the average number of cluster differs greatly for various N. But when the transmission range is about 50m, one node can almost cover the entire network. So it only takes 3 to 5 clusters to cover all the N nodes.



**Fig. 1.** ACN under various transmission range **Fig. 2.** ACN under various maximum velocities

Besides, we do the same research under various velocities. Taking  $N=R=30$  as an example, we can draw the conclusion from figure 2 that: the average number of cluster varies randomly between 5 and 7, and it is not related with the velocity. In fact, it matches with the practical situation too. For example, when one node with large velocity moves out of a cluster, it is highly possible that some other node gets into the same cluster. Or some of the nodes might move toward the same direction, which results in a relatively slow velocity and a stable cluster too.

**5.2 Analysis of Topology Stability**

As is mentioned before, the clusterhead and their affiliated normal nodes may change their roles as they move. Here, we define a cluster reaffiliation factor (CRF) as follows:

$$CRF = \frac{1}{2} \sum_i |N_{i1} - N_{i2}| \tag{2}$$

here,  $i$  is the average number of cluster, and  $N_{i1}, N_{i2}$  are the degree of node  $i$  at different times. For example, we assume that clusterhead 1 and 2 have 6 and 5 neighbors at first, i.e.  $N_{11} = 6, N_{21} = 5$ . As they move after one unit time, their neighbors (degrees) become 5 and 6, i.e.  $N_{12} = 5, N_{22} = 6$ . We can derive that CRF is equal to 1. So, we believe equivalently that one node in cluster 1 moves into cluster 2 and one reaffiliation is made.

Under the maximum velocity of 10 m/s, we compared the CRF performance of Highest-Degree Algorithm, WCA and our GACA. From figure 3, we can see that GACA has the lowest CRF, which shows that it is the stablest clustering strategy among three of them. And WCA has the highest CRF value. The average CRF values of them are 1.56, 0.77 and 0.17 respectively.

Besides, we did some other experiments about CRF. We got the conclusion that the CRF increases as the velocity increases.

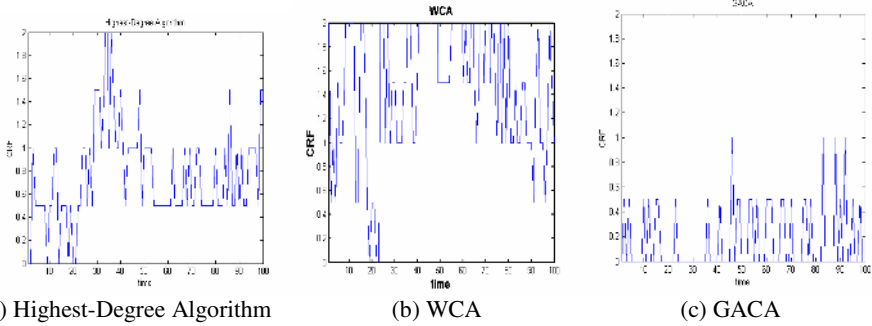


Fig. 3. CRF under various clustering algorithms

5.3 Analysis of Clusterhead Load-Balancing

We take the same definition of load-balancing factor (LBF) as is defined in [8]:

$$LBF = \frac{n_c}{\sum_i (x_i - \mu)^2}, \quad \mu = \frac{N - n_c}{n_c}$$

where,  $n_c$  is the average cluster number,  $N$  is the number of all nodes, and  $x_i$  is the practical degree of node  $i$ . The larger LBF is, the better the load is balanced. Taking

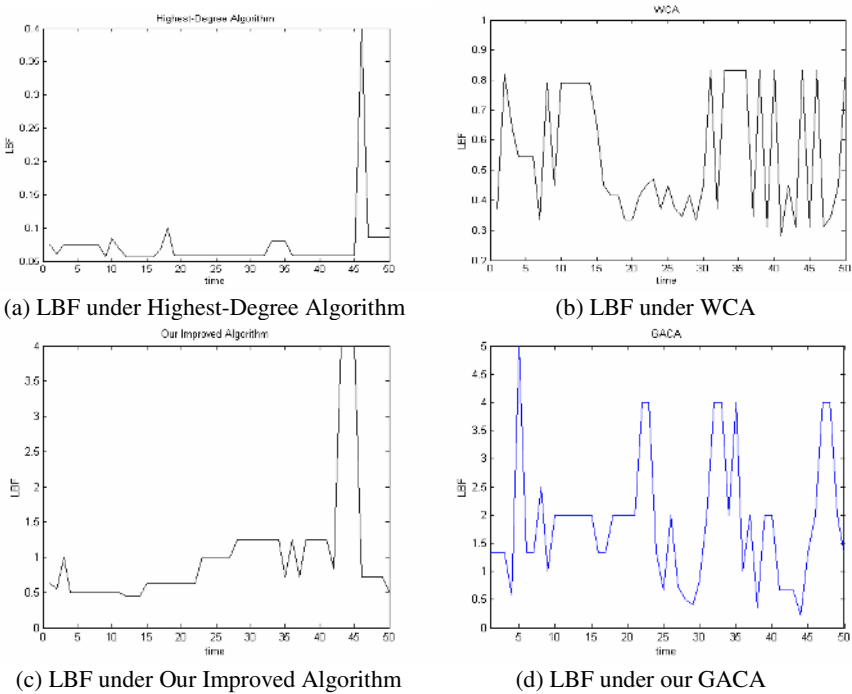


Fig. 4. LBF under various clustering algorithms

$N=20, M=4$  as an example. The ideal case is that there are 4 clusters and each cluster-head has a degree of 4, i.e.  $n_c = x_i = 4$ . Then,  $\mu = (20 - 4) / 4 = 4$ . So LBF is infinite, which shows that the load is perfectly balanced.

For simplicity, we do not consider the factor of network lifetime here (we will discuss it later in next section). So we set the simulation parameters as follows.  $(X,Y)=[100,100]$ ,  $N=20, R=30, M=4$ , maximum velocity  $V_{max} = 5$  and  $w_1 = 0.7, w_2 = 0.2, w_3 = 0.1, w_4 = 0$ . It should be noted that we make  $N_i$  as our primary focus of attention ( $w_1 = 0.7$ ), because it represents the matching degree of the practical case and ideal case directly. Figure 4 shows the LBF distribution under Highest-Degree Algorithm, WCA, our improved weighted clustering algorithm and GACA. From figure 4 we can see that: the Highest-Degree Algorithm has the worst performance, WCA is secondary to it, and our two improved clustering algorithms are better. Besides, the WCA will become useless under densely deployed ad hoc networks while our algorithm still works well. And their average values are 0.09, 0.38, 1.19 and 1.86 respectively.

### 5.4 Analysis of Network Lifetime

Finally, we made a comparison between the aforementioned four clustering algorithms in the aspect of network lifetime, as is shown in figure 5. From which, we can see that GACA achieves the best performance, our improved weighted clustering algorithm is second to it, the WCA and the Highest-Degree algorithm are worse.

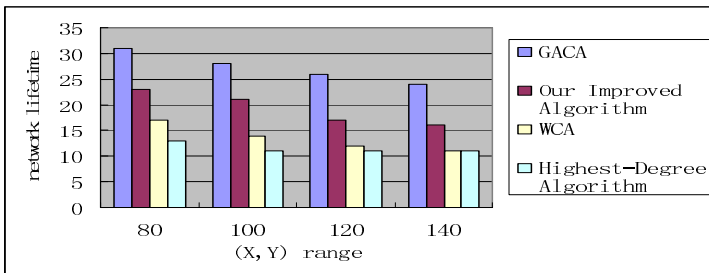


Fig. 5. Network lifetime under various clustering algorithms

## 6 Conclusion

We proposed an improved weighted clustering algorithm based on the WCA and another novel Genetic Annealing based Clustering Algorithm (GACA) in this paper. Some performance comparison is made in the aspect of average cluster number, topology stability, load-balancing and network lifetime. The simulation results show that our two clustering algorithms have a better performance on average.

## Acknowledgement

This work was supported by grant No. R01-2005-000-10267-0 from Korea Science and Engineering Foundation in Ministry of Science and Technology.

## References

1. Internet Engineering Task Force MANET Working Group. Mobile Ad Hoc Network (MANET) Charter [EB/OL]. Available at <http://www.ietf.org/html.charters/manet-charter.html>.
2. C.C. Chiang. Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel [C]. Proceedings of IEEE SICON'97, April 1997, pp.197-211.
3. Mingliang Jiang, Jinyang Li, Y.C. Tay. Cluster Based Routing Protocol [EB/OL]. August, 1999 IETF Draft.
4. A.K. Parekh. Selecting routers in ad-hoc wireless networks [C]. Proceedings of the SBT/IEEE International Telecommunications Symposium, August 1994.
5. M. Gerla and J.T.C. Tsai. Multicluster, mobile, multimedia radio network [J]. ACM/Baltzer Wireless Networks, 1(3),1995, pp. 255-265.
6. D.J. Baker and A. Ephremides. The architectural organization of a mobile radio network via a distributed algorithm [J]. IEEE Transactions on Communicationuns COM-29 11(1981) pp. 1694-1701.
7. S. Basagni. Distributed clustering for ad hoc networks [C]. Proceedings of International Symposium on Parallel Architectures, Algorithms and Networks, June 1999, pp. 310-315.
8. M. Chatterjee, S.K. Das and D. Turgut. An On-Demand Weighted Clustering Algorithm (WCA) for Ad hoc Networks [C]. Proceedings of IEEE GLOBECOM 2000, San Francisco, November 2000, pp.1697-1701.
9. Nishant Gupta, Samir R. Das. Energy-aware On-demand Routing for Mobile Ad Hoc Network. [EB/OL], Available form <http://crewman.uta.edu/~choi/energy.pdf>.
10. MICA2 Mote Datasheet, [http://www.xbow.com/Products/Product\\_pdf\\_files/Wirelesspdf/6020-0042-01\\_A\\_MICA2.pdf](http://www.xbow.com/Products/Product_pdf_files/Wirelesspdf/6020-0042-01_A_MICA2.pdf), 2004.

# Energy-Efficient Cluster Reconfiguration with Fair Cluster Formations in Sensor Networks

Hyang-tack Lee<sup>1</sup>, Yong-hyun Jo<sup>2</sup>, Byeong-hee Roh<sup>1</sup>, and S.W. Yoo<sup>1</sup>

<sup>1</sup> Graduate School of Information and Communication, Ajou University,  
San 5 Wonchon-dong, Yeongtong-Gu, Suwon, 443-749, Korea

{hlee, bhroh, swyoo}@ajou.ac.kr

<sup>2</sup> National Computerization Agency,

NCA building, 77, Mugyo-Dong, Jung-Gu, Seoul, Korea

jyh@nca.or.kr

**Abstract.** In cluster-based schemes such as LEACH, cluster reconfiguration algorithm is one of the most critical issues to achieve longer lifetime of sensor networks. In this paper, we propose a new energy efficient cluster reconfiguration algorithm, called EECRA. EECRA does not require any location or energy information of sensors, and can configure clusters with fair cluster regions such that all the sensors in a sensor network can utilize their energies equally. The performances of EECRA have been compared with LEACH and LEACH-C. We also show that EECRA can be well applied to the environments that sensors are moving.

## 1 Introduction

In microsensor networks, nodes are typically constrained in energy and bandwidth. Therefore, energy-efficient design of the network is one of the most important issues to be solved. There have been much of works related to energy aware routing for sensor networks[1][2][3]. Among those, as distributed cluster-based routing schemes, LEACH (Low-Energy Adaptive Clustering Hierarchy) has been proposed[4]. In LEACH, cluster heads play a key role to aggregate data from sensor nodes in their local cluster regions, and then to deliver the aggregate data to BS. Those cluster heads are periodically elected to prevent a specific sensor node from consuming its residual energy rapidly. However, the cluster heads are elected in distributed and probabilistic way, there exist the possibilities of poor cluster formations, in which cluster heads are located very close to each other. To overcome the problem, LEACH-C (LEACH-Centralized)[5] has been proposed. In LEACH-C, BS(base station) can determine optimum cluster formations based on the information for the locations and residual energies of all sensor nodes. LEACH-C is more effective than LEACH, but it consumes much energy compared with LEACH because all nodes have to communicate with BS at each round and it requires additional overhead for each node to know its location information through an additional communication technique such as GPS.

In this paper, we propose a new energy-efficient cluster reconfiguration algorithm, shortly called EECRA. EECRA uses distributed and probabilistic cluster

formation method as similar as in LEACH. However, EECRA can improve the energy efficiency by preventing the possibility of poor cluster formation as in LEACH. EECRA can produce clusters with fair cluster regions such that all the sensors in a sensor network can utilize their energies equally. The cluster formations configured by EECRA at every round are as optimal as in LEACH-C. Unlike LEACH-C, however, EECRA does not require any location or energy information of each sensor node. Simulation results show that EECRA outperforms both LEACH and LEACH-C.

The rest of the paper is organized as follows. In Section 2, some background on LEACH algorithm and its generic problem is explained. Then, our proposed scheme, EECRA, is illustrated in Section 3. In Section 4, some experimental results will be given. Finally, we conclude the paper in Section 5.

## 2 Background

LEACH is a self-organizing, adaptive clustering protocol that uses randomization to distribute the energy load evenly among the nodes in microsensor network[5]. The basic operation of LEACH is as follows. In LEACH, timeline is divided into rounds as shown in Fig. 1. Each round consists of Set-up Phase and Steady-state Phase. Clusters are reconfigured in Set-up Phase, while actual data transmission can be done from nodes to the cluster head, and then to the BS in Steady-state Phase. Set-up Phase consists of three sub-phases such as Advertisement, Cluster Set-up and Schedule Creation Phases. In Advertisement Phase, each node decides whether it can be elected as a cluster head or not. Then, in Cluster Set-up Phase, all nodes except for cluster heads choose their cluster head, and then cluster reconfiguration is finished. Finally, TDMA schedule for data transmission in the network is arranged in Schedule Creation Phase.

The cluster head election starts at the beginning of each round, especially in Advertisement Phase. Initially, each sensor node chooses a number between 0 and 1 randomly. If the chosen random number by  $n$ -th sensor node is less than a threshold value  $T(n)$  as written in Eq.(1), the node elects itself to a cluster head for the corresponding round.

$$T(n) = \begin{cases} \frac{P}{1-P \cdot (r \bmod \frac{1}{P})} & \text{if } n \in G \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where  $P$  is the desired percentage of the cluster heads,  $r$  is the current round, and  $G$  is the set of nodes that have not been cluster head in the last  $1/P$

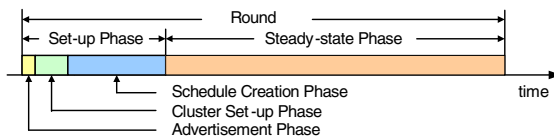
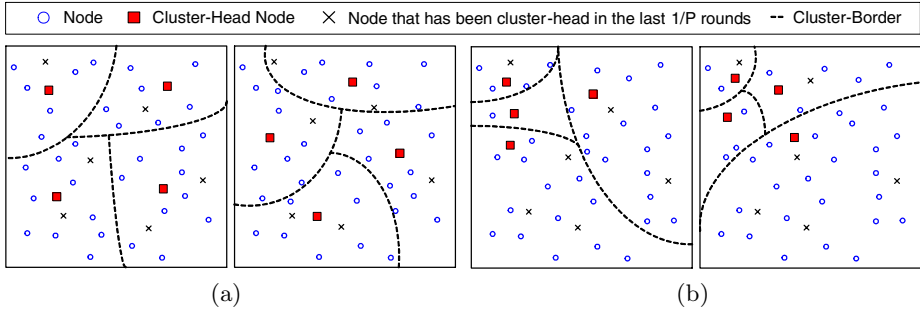


Fig. 1. Timeline of LEACH operation



**Fig. 2.** Possible example of cluster formations: (a) good case (b) poor case

rounds. According to Eq.(1), LEACH ensures that all nodes become a cluster head exactly once during consecutive  $1/P$  rounds.

Because cluster heads are elected in only probabilistic way as shown in Eq.(1), there is no way to consider the formation of clusters in LEACH. Thus, there exists some possibility of both good and poor cluster formations as shown in Fig. 2. In good cluster formations as in Fig. 2(a), all sensor nodes can consume their energy evenly in average. On the other hand, in some poor cluster formations as shown in Fig. 2(b), in which adjacent nodes can be elected as cluster heads, sensor nodes with longer distances to corresponding cluster head consume much more energies than those with shorter distances. In addition, collisions can be occurred frequently in the network due to short distance between cluster heads.

To overcome the problem of poor cluster formation in LEACH, LEACH-C has been proposed[5]. As mentioned in Section 1, however, LEACH-C has also problems of overheads for each node to know its location and deliver its location and energy information to BS. The overhead results in consuming much energy compared with LEACH.

### 3 Energy Efficient Cluster Reconfiguration Algorithm

#### 3.1 Basic Operation of EECRA

Fig. 3 shows the basic timeline of the proposed EECRA operation, in which time-lines are divided into rounds as in LEACH. Unlike LEACH, the Advertisement Phase of EECRA consists of  $k$  stages where  $k$  denotes the predefined number of cluster heads in the network. There exists only one stage in the Advertisement Phase in LEACH. This means that there are  $k$  chances for each node to become a cluster head in EECRA, while only one in LEACH. Let  $N$  be the total number of sensor nodes, and  $T(n, s)$  be the threshold value at  $s$ -th stage that  $n$ -th sensor node can become a cluster head, where  $s=0,1,\dots,k-1$ . Then, we have

$$T(n, s) = \begin{cases} \frac{1}{\{N-k \cdot (r \bmod \frac{N}{k})\} \cdot (1-\frac{s}{k})} & \text{if } n \in G \\ 0 & \text{otherwise} \end{cases} \quad (2)$$



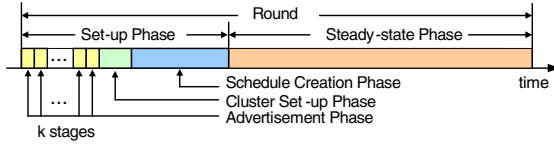


Fig. 3. Timeline of EECRA operation

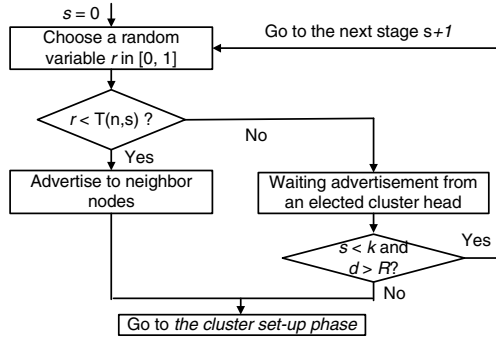


Fig. 4. Cluster head election process of  $n$ -th node

where  $r$  is the current round number. The derivation of Eq.(2) is illustrated in Appendix.

The cluster head election process at each stage is shown in Fig. 4. At the beginning of stage  $s$ , sensor nodes eligible to become a cluster head select a value between 0 and 1 randomly. If the selected random number by a node is less than the threshold  $T(n, s)$ , the node is elected as a cluster head for the corresponding stage. The node elected as a cluster head broadcasts its advertisement message to neighbor nodes within a certain range. Then, those neighbor nodes receiving the advertisement message estimate the distance  $d$  from itself to the cluster head by considering the received message signal power. The nodes with the estimated distance less than  $R$  do not take part in the cluster head election process at the next stage  $s+1$ . We will discuss the value of  $R$  in next subsection. In other words, at the next stage  $s+1$ , only rest nodes that are not belong to the ranges covered by the cluster heads elected at the previous stages can participate in the cluster head election process.

Likewise, EECRA prevents the possibility of unsuitable cluster formations. To do so, EECRA has a little longer Set-up Phase duration than LEACH since it has  $k$  stages. However, we will show in experimental results this can provide more energy-efficient operation of sensor networks than LEACH.

### 3.2 Analysis of R in EECRA

In EECRA, a cluster head elected at each stage broadcast its advertisement message to neighbor nodes within the circle range with radius  $R$ , then nodes

within the range do not take part in head election process at the next stage during corresponding round. Accordingly, to find out suitable value of  $R$  is very important to configure efficient cluster formations. In this paper, we calculate the  $R$  value considering the area of sensor networks.

Consider a sensor network with  $M \times M$  size. Let us assume an ideal situation that sensor nodes are uniformly distributed on the sensor network and  $k$  elected cluster heads are evenly arranged. Then, we have the following approximation

$$M^2 = k\pi R_O^2 \quad (3)$$

where  $R_O$  is the radius for the advertisement region covered by each cluster heads in the above ideal situation. Then, we have  $R_O = \frac{M}{\sqrt{\pi k}}$ . The distance  $R_O$  is suitable only for the ideal situation when the cluster heads are evenly located such that the sum of areas covered by each cluster head is as similar as the area of entire sensor network. However, when some of cluster heads are located on the region near to edge of the network, it becomes unsuitable because the advertisement regions covered by those clusters are smaller than  $M^2/k$ , and it does not satisfy the optimal condition of  $R_O$  as in Eq.(3). To increase the number of nodes that do not need to participate in cluster election process at rest stages and to avoid the poor cluster formation, we can consider some larger values than  $R_O$ . However, we can intuitively expect that the larger the value of  $R_O$ , the higher possibility that cluster nodes at border area can be elected as cluster heads. Since this is not desirable, it needs a certain upper limit of the increase of the radius. Let assume an extreme case when  $k=1$  and the cluster head is elected at the borders of the sensor network. Then, with the radius of  $2R_O$ , the cluster heads can cover the whole sensor network range. So, we can get the upper bound of the radius  $2R_O$ . On the other hand, as the number of cluster heads ( $k$ ) increases, especially let assume that it approaches to infinite, even small area covered by each cluster heads with corresponding small  $R_O$  is enough to satisfy the ideal situation of Eq.(3). Likewise, the radius  $R$  is highly related to the area of sensor networks as well as the number of clusters.

Let  $R_A$  be the average radius of the sensor network. For sensor networks with  $M \times M$  size, it can be approximated by  $M^2 = \pi R_A^2$ . From the intuition that we discussed above, if we define the radius  $R \equiv (1 + \frac{R_O}{R_A})R_O$ , then we have

$$R = (1 + \frac{1}{\sqrt{k}})R_O \quad (4)$$

It is noted that the factor in front of  $R_O$  of right side of Eq.(4) has the range between 1 and 2. This provides the consistency with the intuitions that we used for the derivation of  $R$ . That is,  $R=2R_O$  when  $k=1$ , and  $R \rightarrow R_O$  when  $k \rightarrow \infty$ .

## 4 Experimental Result

### 4.1 Simulation Environment

We carried out the simulations using the Network Simulator ns-2 [6][7]. For the simulation, we consider a sensor network with size of  $100 \times 100$  and 100 sensor

**Table 1.** Basic parameters used for the simulation

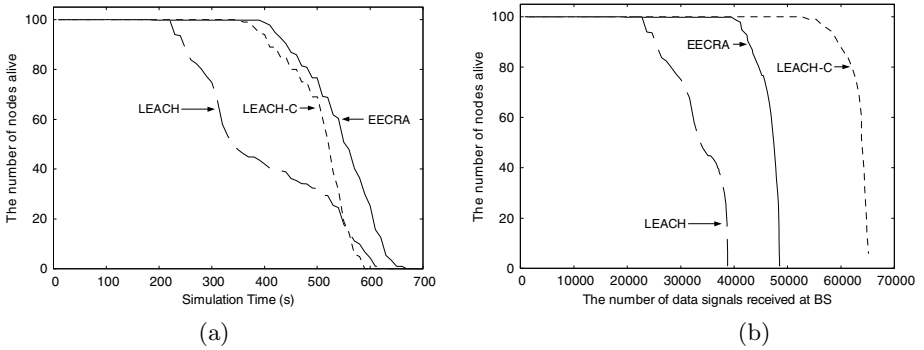
Network Size ( $M \times M$ )	100 $\times$ 100
Location of the nodes	from (0,0) to (99,99)
Location of Base Station	(50,175)
The number of nodes ( $N$ )	100
Desired number of clusters ( $k$ )	5
Initial Energy for each node	2J
Spreading factor	8
Changing clusters	Every 20 seconds

nodes arbitrarily distributed on the sensor network. In our simulation, we use the same simulation environment and radio energy model as discussed in [5]. Our simulation environment is described in Table 1. According to Eq.(4), we selected the radius  $R$  for EECRA as 37.

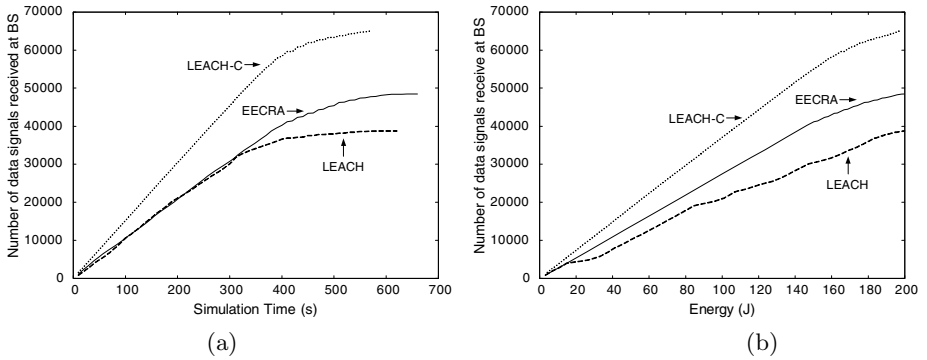
## 4.2 Simulation Results

**In the Case of Static Nodes.** In Fig. 5(a), performances of the system lifetime defined as the time until all nodes are dead are compared. The system lifetime of LEACH-C is shorter than both LEACH and EECRA. This is because in LEACH-C, each node is required to maintain its location and energy information, and has to deliver the information to BS. On the other hand, these overheads are not needed for LEACH and EECRA. So, the energy of each node in LEACH-C is consumed faster than other schemes. And, we can also see from Fig. 5(a) that EECRA has longer lifetime than LEACH. This indicates that EECRA can operate nodes in a very energy-efficient way. Though the system lifetime of LEACH-C is shorter than other schemes, from the viewpoints of the amount of delivered data, LEACH-C outperforms other schemes as shown in Fig. 5(b). This is because LEACH-C can configure optimal cluster formations since BS knows all the operation information of sensor nodes. The data delivery performance of EECRA shows better than that of LEACH. Since LEACH configures clusters based only on probabilistic threshold value, poor cluster formation can be made. Under poor cluster formations, nodes may consume more energy for the same amount of data delivery due to collisions and far distances between some nodes, cluster heads and BS. However, since EECRA forms clusters in consideration of their formations, it can reduce the problem due to poor cluster formations.

In Fig. 6, throughput performances are shown. Fig. 6(a) and (b) present the total amount of data over the time and the total amount of data per given amount of energy received at BS, respectively. As we can expect, BS under LEACH-C receives more amount of data than other schemes because of the optimal cluster formation of LEACH-C with the global knowledge of the network. That is, once clusters are formed in LEACH-C, it requires less energy for data transmission between cluster heads and their cluster member nodes. EECRA shows lower throughput than LEACH-C, but much better than LEACH. This means that EECRA can produce clusters with better formations than LEACH.



**Fig. 5.** Lifetime Performances. (a) number of nodes alive along time (b) number of nodes alive per the amount of data sent to BS.

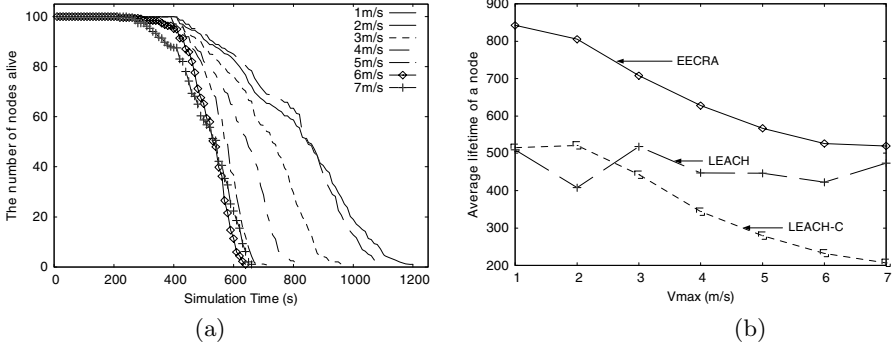


**Fig. 6.** Throughput performances (a) the total number of received data signals at BS over time (b) the total amount of data received at BS per given amount of energy

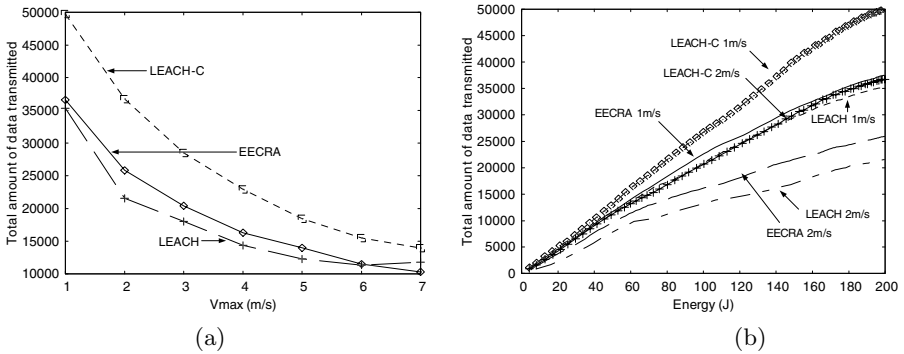
**In the Case of Mobile Nodes.** For the simulation of the environment that sensor nodes are moving, we used Random Waypoint model presented in [8]. At every second, each node randomly chooses a destination and moves toward it with a velocity uniformly chosen from the range  $[0, V_{max}]$ , where  $V_{max}$  is the maximum allowable velocity for every mobile node[9]. The parameters for Random Waypoint model are described in Table 2. We set the values of  $V_{max}$  vary between 1 and 7 m/s to show the efficiency of the network in various mobile environment. In Fig. 7(a), it shows the system lifetime of EECRA over the time for several velocities. As the velocity increases, the number of nodes alive is decreased rapidly along time. We investigated that for other schemes such as LEACH and LEACH-C also show similar pattern as in EECRA of Fig. 7(a). In Fig. 7(b), the lifetimes for EECRA, LEACH and LEACH-C are compared. We can see that EECRA keeps the network longer than LEACH and LEACH-C. And, as in the static node cases, LEACH-C's lifetimes are shorter than LEACH. We can explain the reason of these phenomena as in the static node cases.

**Table 2.** Parameters for Random Waypoint model

$V_{max}$	varies between 1 and 7 m/s
Moving distance per second	$[0 \sim V_{max}]$ m
Direction	$[0 \sim 2]$ Radian
Moving distance per second at x-axis	$[0 \sim V_{max}] \cos([0 \sim 2] \text{ Radian})$ m
Moving distance per second at y-axis	$[0 \sim V_{max}] \sin([0 \sim 2] \text{ Radian})$ m



**Fig. 7.** System lifetime in mobile environments (a) EECRA over time with various velocities (b) comparisons of average node lifetime for three schemes



**Fig. 8.** Throughput in mobile environments (a) total amount of transmitted data (b) total amount of data transmitted per given amount of energy

Fig. 8(a) compares the total number of data signals transmitted during the lifetimes of EECRA, LEACH and LEACH-C. As the velocity increases, the amount of delivered data signals tends to decrease. For all velocities, as we can easily expect, LEACH-C sends more data than EECRA and LEACH. And, EECRA shows always better performance than LEACH. In Fig. 8(b), we show the total amount of data signals transmitted per given amount of energy for those three schemes. LEACH-C always shows the best performances. At 1m/s

velocity, EECRA shows slightly better performances than LEACH. However, at 2m/s velocity, we can see much more performance differences between EECRA and LEACH. Likewise, as velocity increases, EECRA can achieve better performances than LEACH.

Our simulation results show that the system lifetime performance of EECRA outperforms that of other schemes such as LEACH and LEACH-C. However, for the data throughput, LEACH-C shows the best performances, and EECRA is better than LEACH. Though LEACH-C provides the best throughput performances, it has some limitations such as much shorter system lifetime and more complexities to keep the location and energy information in both sensor nodes and BS. From the operational complexity's viewpoints, EECRA can be operated with lower complexity as similar as LEACH, but EECRA provides better performances in both lifetime and throughput than LEACH. In these points of views, EECRA can be more effective than LEACH-C.

## 5 Conclusion

In this paper, we proposed a new energy-efficient cluster reconfiguration algorithm called EECRA. EECRA can reduce the possibility of poor cluster formations appeared in LEACH. With the property, we showed that EECRA can achieve improved lifetime performances than LEACH and LEACH-C under both static and mobile nodes environments. And, EECRA provides reasonable data delivery throughput better than LEACH, but less than LEACH-C. However, EECRA can be implemented less complexity than LEACH-C.

From the performance results of EECRA obtained from experiments, we can conclude that EECRA can be well applied to the situation where longer system operation is required and medium amount of sensing data is requested to be exchanged continuously.

## Appendix: Derivation of Equation (2)

For the derivation of Eq.(2), we let the time at round  $r$  and stage  $s$  be  $(r,s)$  where  $r=0,1,2,\dots$  and  $s=0,1,\dots,k-1$ , and define the following terms at  $(r,s)$ .

$P_n(r,s)$	probability that node $n$ become a cluster head ( $n=1,\dots,N$ )
$C(r,s)$	number of nodes that can take part in cluster head process
$E_{CH}(r,s)$	expected number of nodes that can be elected as cluster heads

From the definition,  $C(r,0)$  is the number of nodes that can be candidates of cluster heads at the beginning of round  $r$ . Then,  $C(r,0) = N - k \cdot (r \bmod \frac{N}{k})$ . In EECRA, it is noted that nodes within the ranges with radius  $R$  from cluster heads elected at every stages can not take part in the cluster head process at the next stage. The average number of nodes belongs to the range covered by each cluster head can be  $C(r,0)/k$ . Accordingly, we can write

$$C(r,s) = C(r,0) \cdot \left(1 - \frac{s}{k}\right) \quad (\text{A1})$$

Ideally, one cluster head is elected at each stage. So, the probability  $P_n(r, s)$  satisfies the following condition.

$$E_{CH}(r, s) = \sum_{n=1}^N P_n(r, s) = 1 \quad (\text{A2})$$

Let  $G$  be the set of nodes that have not been elected as cluster heads in the last  $N/k$  rounds, and  $P_{n \in G}(r, s)$  be the probability that an arbitrary node  $n$  ( $n \in G$ ) becomes a cluster head at time  $(r, s)$ . The nodes that have not been cluster head have identical  $P_n(\cdot)$  at each stage. Then, we can rewrite Eq.(A2) as following

$$E_{CH}(r, s) = C(r, s) \cdot P_{n \in G}(r, s) = 1 \quad (\text{A3})$$

From Eq.(A3), we have

$$P_{n \in G}(r, s) = \frac{1}{C(r, s)} \quad (\text{A4})$$

It is noted that Eq.(A4) is the probability that only one node can be elected as the cluster head at a certain stage. Substituting Eq.(A1) into Eq.(A4), we have the equation as same as Eq.(2).

**Acknowledgements.** This work was supported by grant (No. 05A3-I3-10) from Ubiquitous Autonomic Computing and Network Project sponsored by the Ministry of Information and Communication, Korea.

## References

1. Edger H. Callaway, *Wireless Sensor Networks: Architectures and Protocols*, Auerbach Publications, August 2003
2. Sameer Tilak, Nael B. Abu-Ghazaleh, and Wendi Heinzelman, "A Taxonomy of Wireless Micro-Sensor Network Models," *ACM Mobile Computing and Communications Review*, Vol. 6, No. 2, pp. 28-36, April 2002
3. I.F.Akyildiz, W.Su, Y.Sankarasubramaniam, E.Cayirci, "A survey on Sensor Networks," *IEEE Communications Magazine*, Vol. 40, No. 8, pp. 102-114, August 2002
4. W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," *Proceeding of Hawaii Conference on System Sciences*, January 2000
5. W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," *IEEE Transactions on Wireless Communications*, Vol. 1, No. 4, October 2002
6. UCB/LBNL/VINT, "Network Simulator ns-2," <http://www.mash.cs.berkeley.edu/ns>.
7. W.Heinzelman, A.Chandrakasan, and H.Balakrishnan, uAMPS ns Code Extensions, <http://www.mtl.mit.edu/research/icsystems/uamps/leach>
8. F.Bai, N.Sadagopan, A.Helmy, "The IMPORTANT Framework for Analyzing the impact of Mobility on Performance of Routing for Ad Hoc Networks," *Ad-Hoc Networks*, Vol. 1, No. 5, pp. 383 - 404, November 2003
9. L. Breslau, D. Estrin, K. Fall, S. Floyd, J. Heidemann, A. Helmy, P. Huang, S. McCanne, K. Varadhan, Y. Xu, H. Yu, "Advances in network simulation," *IEEE Computer Magazine*, Vol. 33, No. 5, pp. 59-67, May 2000

# Virtual Sink Rotation: Low-Energy Scalable Routing Protocol for Ubiquitous Sensor Networks\*

Lynn Choi<sup>1</sup>, Kwangseok Choi<sup>1</sup>, Jungsun Kim<sup>2</sup>, and Byung Joon Park<sup>3</sup>

<sup>1</sup> Korea University, Anam-Dong, Sungbuk-Ku, Seoul, Korea  
{lchoi, jsheo, hyogon}@korea.ac.kr  
Tel: +82-2-3290-3249, Fax: +82-2-921-0544

<sup>2</sup> School of Electrical Engineering and Computer Science, Hanyang University  
jskim@cse.hanyang.ac.kr

<sup>3</sup> Department of Computer Science, Kwangwoon University  
bjpark@cs.kwangwoon.ac.kr

**Abstract.** In this paper we propose a new routing protocol called virtual sink rotation (VSR) routing for large-scale sensor networks. VSR can efficiently handle a large number of sources as well as a large number of sinks with potential mobility. Each sensor node is not required to know the global network topology nor the location awareness. The main ideas underlying the VSR are two folds. First, to alleviate the frequent location updates associated with multiple mobile sinks, the algorithm introduces a *virtual sink*, which acts as a data collection and dissemination center to collect the data from all the sources and forward them to the actual sinks. This virtual sink can easily support multiple sinks as well as the mobility of the sinks. Second, to address the excessive energy consumption among the sensor nodes around a sink, VSR employs *virtual sink rotation*, which distributes the role of the virtual sink over all the participating sensor nodes, thus achieving a uniform energy distribution across the entire sensor field and prolonging the lifetime of the network. Experimentation results confirm that the VSR routing can significantly save energy while it can also reduce both the message delay and the message delivery failures compared to previous schemes.

## 1 Introduction

Sensor network routing protocols for stationary sinks has been well studied [1, 2, 3, 6, 7, 8, 10, 11, 13]. However, the routing protocols for mobile sinks still remain an open area, where no practical solution has been discovered. Sink mobility brings new challenges to large-scale sensor networking. First, the location information of a mobile sink needs to be continuously propagated throughout the sensor field to keep all sensor nodes updated with the location to send future data reports. Unfortunately the frequent location updates from a mobile sink can lead

---

\* This research was supported by University IT Research Center Project under contract number C1090-0401-0014.



to excessive drain of sensors' limited battery power supply and increased collisions in wireless transmissions. Second, the location of a mobile sink can be lost since often the location update may not be fast enough. In such cases a message sent from a source may not be delivered to the mobile sink. Third, the preconstruction of message delivery network or tree cannot be performed since the sink may move and the existing tree may not help immediately after the movement. Fourth, this situation becomes exacerbated when the number of such mobile sinks grows.

Although several sensor network routing protocols have been proposed recently to target mobile sinks, such as TTDD [11], SEAD [7], and HLETDR [2], most of them suggest that each mobile sink needs to continuously propagate its location information throughout the sensor field, either by a local flooding guided by geographical grids pre-maintained [11] or by a global flooding based on localized interactions among the neighboring nodes through interest propagation and reinforcement [2, 6]. Moreover, some of existing schemes assume location awareness using GPS-enabled nodes [7, 11]. However, GPS may not be used in many wireless sensor networks as GPS can work only outdoors with no obstruction and the cost of GPS receivers prevents them from using in low-cost sensor nodes. Furthermore, most of the proposed localization techniques today depend on recursive trilateration/multilateration techniques, which would not provide enough accuracy in wireless sensor networks [4]. Thus, none of the existing approaches provides an efficient and practical solution to this problem.

In this paper we propose a new routing protocol called virtual sink rotation (VSR) routing for large-scale sensor networks. VSR can efficiently handle a large number of sources as well as a large number of sinks with potential mobility. Each sensor node is assumed to be GPS-free, thus suitable for low-cost sensor node implementation. In addition, each node maintains information about its neighbors, thus no global network topology information needs to be maintained. This enables to build low-cost scalable routing protocols for large-scale sensor networks.

The main ideas underlying the VSR are two folds. First, to alleviate the frequent location updates associated with multiple mobile sinks, the algorithm introduces a *virtual sink*, which acts as a data collection and dissemination center for all the sensor nodes. The virtual sink builds a spanning tree encompassing all the sensor nodes in the field to collect and sometimes to aggregate the data from all the sources. Then, the virtual sink builds a multi-cast tree to the actual sinks to disseminate the data collected from the sources. This virtual sink can easily support multiple sinks as well as the mobility of the sinks since each sensor node does not need to keep track of sink's location information any more. Only the virtual sink needs to keep track of mobile sinks' locations. Second, to address the excessive energy consumption in the sensor field around a sink, VSR employs *virtual sink rotation*, which distributes the role of the virtual sink over all the sensor nodes participating. This enables a uniform distribution of energy consumption across the entire sensor field so that the lifetime of the sensor network can be extended. Thus, the excessive energy drain of a particular area does neither cause network partitioning nor it creates the holes in the sensor field coverage. Experimentation results confirm that the VSR routing can significantly save energy while it can also reduce both the message delay and the message delivery failures compared to previous schemes.

## 2 Related Works

While most of existing routing protocols implicitly assume sensor networks with stationary sinks or sinks with very low mobility, TTDD [11] and SEAD [7] specifically target sensor networks with multiple mobile sinks. In TTDD each data source proactively builds a grid structure, which enables mobile sinks to continuously receive data on the move by flooding queries within a local cell only. TTDD's design exploits the fact that sensor nodes are stationary and location-aware to construct and maintain the grid structures with low overhead. A more recent scheme called SEAD resembles VSR in the sense that SEAD does not rely on local or global flooding. Instead, SEAD builds an overlay multicast network called d-tree from each source to multiple sinks. To minimize data dissemination cost, source data is replicated at selected nodes between the source and the sinks. SEAD is different from VSR in that each sensor node in SEAD is assumed to be aware of its geographical location. In addition, the d-tree in SEAD is not a spanning tree but an overlay network connecting a single source and multiple sinks on top of underlying location-based routing protocols such as simple geographical forwarding [10, 13].

The rest of this paper is organized as follows. In Section 3 we present the concepts of VSR routing, namely the virtual sink and virtual sink rotation, and describes its relevant algorithms required for its spanning tree construction, virtual path setup and removal, and virtual sink selection policies. In Section 4 we present our simulation methodology and the evaluation results of VSR compared to existing schemes. Section 5 concludes the paper and describes our future work.

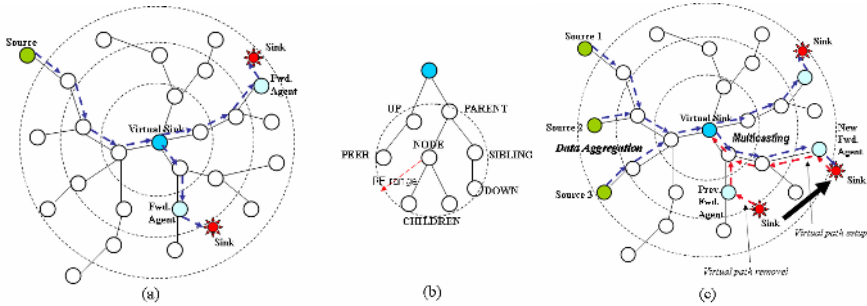
## 3 Virtual Sink Rotation (VSR) Routing

### 3.1 Virtual Sink

Virtual Sink Rotation (VSR) routing algorithm is based on two important concepts: virtual sink and virtual sink rotation. In this section we first describe the concept of virtual sink and its associated algorithms.

**Virtual Sink.** A *virtual sink* is a sensor node that is delegated to perform the role of all the actual sinks. A sensor node selected as a virtual sink is authorized to act as a data collection center for all the sources and also as a data dissemination center for all the sinks. To perform the data collection and dissemination efficiently, the virtual sink builds a spanning tree called *VS tree* that encompasses all the sensor nodes in the field to collect and potentially aggregate the data from all the sources. The VS tree is constructed at the initial deployment of sensor nodes and can be repaired or reconstructed from time to time due to node failures, excessive energy drains surrounding the virtual sink, or the additional deployment of new sensor nodes. Once the VS tree is constructed, the virtual sink can act as a data dissemination center for sinks. To forward the data sent from a source, the virtual sink builds a data dissemination path from the source to the actual sink through the virtual sink. When multiple sinks are present, multiple data dissemination paths from the virtual sink to the actual sinks can naturally form a dissemination multi-cast tree to disseminate the data collected from the

sources to the actual sinks. This virtual sink can easily support multiple sinks as well as the mobility of the sinks since each sensor node does not need to keep track of sink's location information anymore. Only the virtual sink needs to keep track of mobile sinks' locations. And, the location update from an actual sink to the virtual sink can be easily performed using the existing path in the VS tree without local or global forwarding required by previous schemes [2, 6, 11].



**Fig. 1.** (a) An example of VS tree consisting of 26 sensor nodes organized into 4 layers, (b) classification of neighbor nodes, (c) an illustration of a virtual path removal and a new virtual path setup on sink movement

**VS Tree Construction.** VSR organizes the sensor network into layers, where nodes that belong to a layer have the same hop-count to the virtual sink. Thus, layer 1 consists of nodes which can reach the virtual sink in one hop; layer 2 nodes reach the virtual sink in two hops, and so on. Figure 1(a) illustrates how 26 sensor nodes are organized into four layers. Thus, all the sensor nodes in the sensor field constitute a spanning tree rooted at the virtual sink. The VS tree construction is initiated by a *tree setup message* sent by the virtual sink. This tree setup message contains its own id, its parent node id, and the number of hops to the virtual sink. All the nodes hearing the message from the virtual sink become the children of the virtual sink and constitute the layer 1 nodes. After hearing this message, nodes in the layer 1 rebroadcast the tree setup message to reach the layer 2 nodes. By overhearing these tree setup messages from a layer below, the virtual sink can record its children nodes that it can reach in a single hop. This process is repeated until all the sensor nodes can be reached by the tree setup message. Using the broadcast nature of wireless transmissions, the VS tree can be easily constructed from the root downward and each node is required to send the tree setup message only once. To reduce the communication latency between a sensor node and the virtual sink, the height of this tree must be minimized since the height of a node represents the number of hops to reach the virtual sink. And, the result tree must span all the sensor nodes in the field. In this regard, the VS tree construction algorithm is to build a spanning tree with a minimum height.

In a VS tree neighbors of a node can be classified as either UP, DOWN, or PEER depending on their proximity to the virtual sink. A node  $i$  is considered as UP with respect to a node  $j$  if  $H(i) < H(j)$ , where  $H(i)$  denotes the height of a node  $i$ . Similarly a node  $i$  is considered DOWN with respect to a node  $j$  if  $H(i) > H(j)$ . Finally a node  $i$  is considered PEER with respect to node  $j$  if  $H(i) = H(j)$ . Each node should have at least

one UP node in its neighbors to disseminate its data to the virtual sink. Each node selects one in UP neighbors as a PARENT node and it becomes the CHILD node of the PARENT. In addition, a node may or may not have DOWN neighbors depending on its location in the VS tree. To forward the data sent from the virtual sink to an actual sink, each node in this path needs to forward the data coming from its UP neighbor to one of its DOWN nodes. Thus, both UP and DOWN links need to be maintained for the data dissemination. In summary, each node has at least one PARENT node and zero or more CHILDREN nodes. And, its all other neighboring nodes are classified into UP, DOWN, and PEER as illustrated in Figure 1(b).

**Virtual Path.** Once the VS tree is constructed, the virtual sink can function as a data collection center for all the sources in the field. Each sensor node has a path to the virtual sink through its PARENT node upward in the VS tree and this upward path is called its *report path*. Now to disseminate the data sent from the sources, the virtual sink needs to build a path downward to real sinks. This downward path is called *virtual path*. The upward path in connection with the downward path constitutes the data dissemination path from a source to an actual sink. To establish the virtual path to each real sink, the virtual sink must be informed of the location of a real sink. This process is called *virtual path setup* and initiated by each real sink. Since every sensor node in VS tree already knows the path to the virtual sink, the virtual path setup can be easily performed in VSR using the existing report path.

**Virtual Path Setup and Removal.** In VSR, each mobile sink is associated with one stationary sensor node called *forwarding agent*. When a new sink enters the sensor field, the sink selects one of its neighboring sensor nodes that has the smallest hop count to the virtual sink and sends the *virtual path setup message* to the neighbor node. The selected sensor node is called the sink's *forwarding agent*. Since this forwarding agent already knows its report path to the virtual sink, the virtual path can be easily set up using this existing path from the agent to the virtual sink. Since the direction of data movement in the virtual path, i.e. downward from the virtual sink to the real sink, is the opposite of the report path, this setup message is forwarded upward to the virtual sink through the report path of the forwarding agent. On its way upward, each node in the virtual path records the direction of a new sink. When the virtual path setup message reaches the virtual sink, the virtual sink can now record the direction of the new sink. Finally, the virtual sink can function as a data dissemination center for the new sink. This way the data dissemination path from all the sources to the particular sink can be established. The virtual path setup is also performed when a sink moves or when the VS tree is reconstructed.

When a sink is about to move out of the range of its current forwarding agent, the existing virtual path to the sink is no longer valid. During this sink movement, the data sent from sources can be lost if the setup of a new virtual path is not fast enough. To avoid such message delivery failures, VSR employs *virtual path removal*. That is, the mobile sink informs its forwarding agent of its movement and lets its agent to invalidate the old virtual path by sending virtual path removal message. This message must be propagated to the virtual sink through the old virtual path. If a node in the old virtual path already has the data to send to the mobile sink, this data can be sent back to the virtual sink that can temporarily buffer all the messages toward the mobile sink.

Once the new location of the mobile sink is stabilized, the sink must set up the virtual path again by selecting one of its neighbors as a new forwarding agent and sending the virtual path setup message to this agent. When this message reaches the virtual sink, the virtual sink can now forward all the data to the mobile sink including the buffered messages during the mobile sink's movement. Using the virtual path setup and removal, VSR can successfully forward messages for the mobile sink without losing messages and without indirect forwarding between the agents [11]. As we demonstrate in Section 3, this virtual path setup can be performed very efficiently in terms of energy, delay, and the rate of success deliveries.

**Dissemination Multicast Tree.** When the number of sinks grows, virtual paths from multiple sinks can be naturally grouped into a multi-cast tree without any additional tree construction overhead. This is because every virtual path is already embedded in the VS tree. When two virtual paths join at a node in the VS tree, two virtual paths can be naturally combined from the join node upward to the virtual sink. The result tree starts from the virtual sink and ends at the two mobile sinks. When the number of sinks grows, the tree naturally expands. The root of the tree is the virtual sink whereas the leaves of the tree are the mobile sinks. We call this tree *dissemination multicast tree*, which can be used to multicast data more efficiently. Since each virtual path is embedded in the VS tree, the result multi-cast tree can easily exploit the spatial locality among the mobile sinks by utilizing the path proximity among the virtual paths. Both the virtual path and the dissemination multi-cast tree are embedded in the VS tree as illustrated in Figure 1(c). With the dissemination multicast tree, the virtual sink can now act as a data dissemination center for all the sinks in the field.

Note that all the communication in VSR is strictly local, i.e. neighbor-to-neighbor. That is to say, every sensor node including the virtual sink needs to keep track of only its neighbors. This suggests that VSR has two desirable features. First, a sensor node in VSR is not assumed to be aware of its location, i.e. does not rely on GPS to implement the routing protocol. Thus, VSR can be easily employed in low-cost GPS-free sensor node implementations. Second, a sensor node does not have to know the global topology of the network, which implies that each node only keeps track of information about neighbor nodes in its routing table. Thus, the VSR is scalable in terms of the number of nodes in the network since the amount of information kept in each node is constant even though the number of nodes in the field grows.

### 3.2 Virtual Sink Rotation

One of the key issues in low-energy sensor network protocol design is the excessive energy drain among the nodes around a sink. Since a sink usually communicates with a number of sources, this type of many-to-one communication causes heavy traffic around the sink. Thus, interior nodes nearby the sink usually experience more energy consumption than exterior nodes in the field. The same phenomenon occurs among the nodes around the virtual sink in VSR. The situation can be even worse since VSR assumes a normal sensor node as the virtual sink whereas existing routing protocols assume a more powerful sink, which is different from a normal sensor node. To get around this problem, many energy-aware routing protocols have been proposed to distribute the energy consumption throughout the sensor field [3, 8]. However, all of them so far suggest local optimization for this problem in the sense that the sink and

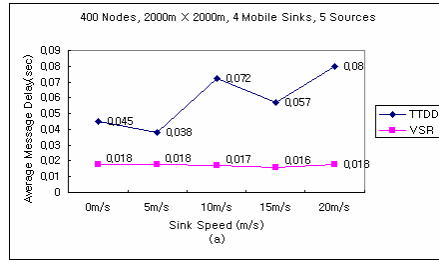
its immediate neighbors cannot avoid excessive energy drain. Instead, we take a global optimization approach called *virtual sink rotation*. In virtual sink rotation, when the energy level of the virtual sink reaches a certain threshold, VSR selects another node as a new virtual sink to evenly distribute the energy consumption throughout the sensor field.

**Virtual Sink Selection Strategy.** If there is only one sink in the field, one of the sink's neighbors can be selected as the virtual sink. When a new sink joins the sensor field, the location of the virtual sink may be far away from this sink depending on its relative position to the previous sink. To minimize energy, the location of the virtual sink must be chosen to minimize the overall communication distance between the virtual sink and multiple sinks. Thus, the optimal position of the virtual sink may be the location corresponding to the weighted center of all the sinks' locations assuming that the sources are evenly distributed throughout the sensor field. However, since each sink is free to move anytime in our network model, we do not take this approach. Instead, we employ global optimization technique called virtual sink rotation, where the role of the virtual sink is rotated among the sensor nodes to distribute the energy as evenly as possible. For the virtual sink selection, we take a simple approach called *random selection*, which randomly picks one sensor node as the virtual sink. We expect this strategy to perform well compared to a more sophisticated approach assuming that the number of sensor nodes in the field is sufficiently large.

## 4 Experimentation and Results

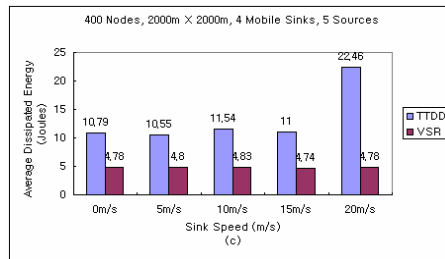
We implemented the VSR protocol in the ns-2 simulator [9]. The ns-2 simulator implements 1.6Mbps 802.11 MAC layer. This MAC layer may not be suitable for sensor networks since it is possible to put the radio in standby mode during idle intervals as in other sensor network MAC protocols [5, 12]. By contrast, an 802.11 radio consumes as much power when it is idle as when it receives transmissions. To more closely model realistic sensor network model, we altered the ns-2 radio energy model such that the idle-time power dissipation is about 35mW, reception power dissipation of 395mW, and transmission power dissipation of 660mW. This is consistent with previous sensor network studies we compared.

All the simulations data are collected by varying the speed of each mobile sink from 0 to 20 m/s in the sensor field of 400 nodes, where the nodes are randomly placed in a 2000m  $\times$  2000m square. Five sources and four mobile sinks are used in each simulation run. Each node has a radio range of 250m. Unless otherwise mentioned, all sources are randomly selected from the sensor field following the random sources model [6] while sinks are uniformly scattered across the field. Each source generates one event per second and each event is modeled as a 64-byte packet. All the events are reported to all the sinks in the field. Each simulation run lasts for 100 seconds. All the metrics of VSR are compared against only Two-Tier Data Dissemination (TTDD) scheme, which is a protocol specifically designed to handle multiple mobile sinks.



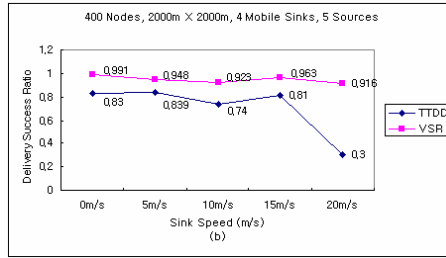
**Fig. 2.** Average message delay for different speeds of mobile sinks

Figure 2 shows the average message delay for different speeds of mobile sinks. As the speed of mobile sinks increases, VSR's average message delay remains stable. This is because VSR only incurs one virtual path setup for each mobile sink regardless of its speed. By contrast, TTDD requires the former primary agent to forward all the messages to the new immediate agent during the sink movement, which increases the message delay. Moreover, TTDD often results in local flooding to select a new primary agent if the sink moves out of a certain distance from its primary agent. This local flooding increases the contention in the network, further delaying the message delivery during the fast movement of the sinks. However, in VSR the average message delay remains almost the same regardless of the sink speed. Furthermore, the average message delay remains below 20ms, which is at least twice faster than TTDD for all the cases simulated.



**Fig. 3.** Average dissipated energy for different speeds of mobile sinks

Figure 3 shows the average dissipated energy. For all different speeds of mobile sinks VSR's average dissipated energy remains relatively constant, around 5J. Similarly, the average dissipated energy of TTDD remains stable at 11J for low to medium speed sink movement. However, TTDD's per-node energy consumption jumps to 22.4J when the speed of mobile sink reaches 20m/s due to its excessive local flooding and trajectory forwarding caused by the fast movement. Thus, in terms of both energy and delay, VSR substantially outperforms TTDD.



**Fig. 4.** Delivery success ratio for different speeds of mobile sinks

Lastly, Figure 4 shows the delivery success ratio. As we expect from low delay and low energy consumption of VSR, VSR consistently delivers more than 91% of all the events to the mobile sinks for all different speeds, whereas TTDD's delivery success ratio remains around 80% when the speed of mobile sinks ranges from 5m/s to 15m/s. However, the TTDD's delivery success ratio suddenly drops to 30% when the speed of mobile sinks reaches 20m/s, which suggests that TTDD's protocol overhead cannot sustain fast movement of mobile sink beyond this speed.

## 5 Conclusion

This paper addresses the problem of scalable and efficient data dissemination in a large-scale sensor network from multiple sources to multiple, potentially *mobile* sinks. Although routing protocols for sensor networks have been an active research field in the past few years, most of them target sensor networks with immobile sinks or sinks with low mobility. Sink mobility brings new challenges to sensor networking since frequent location updates from mobile sinks often result in unnecessary traffic, which increases the message delay as well as the energy dissipation per node.

In this paper we propose a new routing protocol called virtual sink rotation (VSR) routing for sensor networks. VSR introduces a virtual sink, which acts as a data collection and dissemination center for all the sensor nodes in the field. In addition, VSR employs virtual sink rotation, which distributes the role of the virtual sink over all the participating sensor nodes, thus distributing the energy consumption across the entire sensor field. We have implemented VSR in ns-2 and evaluated the performance of VSR by comparing it to two previous schemes in terms of average message delay, average dissipated energy, and delivery success ratio. Experimentation results confirm that VSR can significantly save energy while it can also reduce both the message delay and the message delivery failures compared to previous schemes. Moreover, each sensor node in VSR is not assumed to know the global network topology nor its position, suggesting that VSR can be a scalable routing solution for large-scale sensor networks implemented with low-cost sensor nodes and mobile sinks.

In our future work, we will investigate several new issues including the impact of data and query aggregation, caching, and the multicasting on the VSR framework.



## References

1. Al-Karaki, J. N. and Kamal, A. E., Routing Techniques in Wireless Sensor Networks: A Survey, *Wireless Communications, IEEE*, pp. 6-28, Vol. 11, Issue: 6, Dec. 2004.
2. Baruah, P., Urgaonkar, R., and Krishnamachari, B., Learning-Enforced Time Domain Routing to Mobile Sinks in Wireless Sensor Fields, *Local Computer Networks*, pp. 525 – 532, 2004.
3. Boukerche, A., Cheng, X., Linus, J., Energy-aware Data-centric Routing in Microsensor Networks , In *Proceedings of the 6<sup>th</sup> International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems*, pp. 42 - 49 , Sep. 2003.
4. Bulusu, N., Estrin, D., Girod, L. and Heidemann, J., Scalable Coordination for Wireless Sensor Networks: Self-Configuring Localization Systems", In *Proceedings of the Sixth International Symposium on Communication Theory and Applications (ISCTA 2001)*, July 2001.
5. Christian C. Enz., Amre, E. H., Decotignie, J. D., Peiris, V., WiseNET: An Ultralow-Power Wireless Sensor Network Solution, *IEEE computer magazine*, Vol. 37, No. 8, Aug 2004.
6. Intanagonwiwat, C., Govindan, R. and Estrin, D., Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks , In *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networks*, pp. 2 – 16, Vol. 11, Issue 1, 2000.
7. Kim, H. S., Abdelzahan, T. F., Kwon, W. H., Minimum-Energy Asynchronous Dissemination to Mobile Sinks in Wireless Sensor Networks, In *Proceedings of the First International Conference on Embedded Networked Sensor Systems*, pp. 193 – 204, Nov. 2003.
8. Lindsey, S. and Raghavendra, C. S., Pegasus : Power-Efficient Gathering in Sensor Information Systems, In *Proceedings of IEEE international Conference on Communications*, pp. 1125 -1130, Vol.3, June 2001.
9. The Network Simulator ns-2 Documentation, <http://www.isi.edu/nsman/ns>.
10. Xu, Y., Heidemann, J., Estrin, D., Geography-Informed Energy Conservation for Ad Hoc Routing , In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, pp. 70 - 84, 2001.
11. Ye, F., Luo, H., Cheng J., Lu, S., Zhang, L., A Two-Tier Data Dissemination Model for Large-Scale Wireless Sensor Networks, In *Proceeding of Mobile Computing and Networks*, pp. 148 – 159, Sep. 2002.
12. Ye, W., Heidemann, J., Estrin, D., An Energy-Efficient MAC Protocol for Wireless Sensor Networks, In *Proceedings of the INFOCOM 2002*, pp. 1567 – 1576, Vol. 3, 2002.
13. Yu, Y., Govindan, R. and Estrin, D., Geographical and Energy Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks, *UCLA Computer Science Department Technical Report UCLA-CSD TR-01-0023*, 2001.

# FERMA: An Efficient Geocasting Protocol for Wireless Sensor Networks with Multiple Target Regions<sup>★</sup>

Young-Mi Song, Sung-Hee Lee, and Young-Bae Ko

College of Information and Communication,  
Ajou University, Suwon, South Korea  
{ymsong, sunghee, youngko}@ajou.ac.kr

**Abstract.** Some sensor applications are interested in collecting data from multiple regions. For supporting such applications with multiple target regions, most conventional protocols are based on either a network flooding or multiple unicastig to cover those more than one target region. Either one will result in a lot of redundant packets to transmit by energy scared sensor nodes. To alleviate this problem, we propose a novel geocasting scheme which can make a suitable shared path among multiple target regions. We utilize the theorem of “Fermat Point,” in order to find an optimal junction point branching into each region. By using this shared path, an interest dissemination can be performed very efficiently. Our simulation study shows that the proposed scheme FERMA reduces a lot of network traffic and achieves significant energy saving as the number of target regions increase.

## 1 Introduction

Advances in wireless embedded technologies make it possible to enable small and resource-limited sensor devices to have wireless communication and computational capabilities [1]. Wireless sensor networks (WSNs) with a large number of such smart sensors can be deployed for tracking targets or gathering information about physical phenomena. For many applications in wireless sensor networks [2, 3], a query (also, called as an *interest*) is commonly used to have sensor nodes collect data from their environments and return these sensing data to a query initiator (i.e., the originator of the interest message). In such query-based sensor networks, an interest message specifies a particular condition to match events; for example, a type of sensing tasks, location information where interesting events might occur, an interval between data propagations, and the lifetime of the query.

There are several approaches for efficiently disseminating interest messages to a target region. A flooding mechanism, which requires any intermediate receiver to rebroadcast a non-duplicated interest packet to all its neighbors, is the most commonly used technique. For example in directed diffusion [3], one of the well-known query-based routing protocols in wireless sensor networks, a sink node is

---

<sup>★</sup> This work was supported by the Korea Research Foundation Grant funded by the Korean Government (R05-2003-000-10607-02004) and also supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC program.

supposed to initiate an interest packet dissemination throughout the entire network by flooding. Then a node receiving the interest sets up a *gradient* which indicates from whom this interest message has previously been forwarded. Although some additional feature such as a gradient reinforcement has been proposed, the directed diffusion with such a flooding of interest messages obviously increases network traffic and leads inefficient energy consumption on sensor nodes.

A geographical propagation can be more useful than flooding when an interest message needs to send towards a subset of sensor need within a certain region. Thus, a geocasting can reduce the number of redundant packets by aid of the location information of nodes. GEAR [4] is one of geocasting protocols for sensor networks. It utilizes a greedy forwarding for the packet delivery toward the target region. In greedy forwarding, a packet is forwarded to only one of the neighbor nodes whose geographical location is closest to the destination. Therefore, GEAR can minimize redundant packet traffic caused in flooding. However, most conventional geocasting protocols including GEAR consider only a single target region. The problem we are addressing in this paper is how to efficiently deliver interest messages to multiple target regions so that the latency as well as the bandwidth consumption can be reduced.

There are many sensor network applications, which require to collect the identical information from multiple regions. For example, a monitoring system in hostile environments may need to send the same advertisement to the sensors in the several regions for changing the sensing mode or interval. It may also need to send some queries such as “what’s the average temperature in each target region A, B and C?”. Additionally, in the battle field, the command center can give the same query to the sensors within multiple combat areas. Example queries for such scenarios would be, “How many tanks or soldiers are observed in regions X, Y and Z?” or “Where are tanks in regions X, Y and Z?”. For these applications, conventional protocols need to send the identical interests to each target region multiple times. It causes significant performance degradation by increasing network traffic and wasting the energy.

We propose a noble scheme that sends interest just once at the sink node instead of multiple packet transmissions toward different target regions. Our scheme, named FERMA, creates a suitable shared path among multiple target regions. The interest messages are then forwarded along this path from the sink node to each target region through an optimal junction point. To find such an optimal junction point, we utilize the theorem of the “Fermat Point” [5]. After the interest reaches any one node in each target region, local flooding starts inside the region. A gradient, which represents the reverse path of the interest, is set up toward the sink node in the process of the interest forwarding. This gradient is utilized for actual data deliveries from sensor nodes to the sink node.

Our scheme also performs a data aggregation to reduce the amount of data traffic. It is done at each Fermat Point and entrance nodes of local flooding inside the target regions. The rest of the paper is organized as follows. Section 2 introduces the theorem of Fermat Point and its proof. In Section 3, we examine our proposed scheme in aspects of interest forwarding and data forwarding followed by ns-2 simulation results in Section 4. We conclude in Section 5.

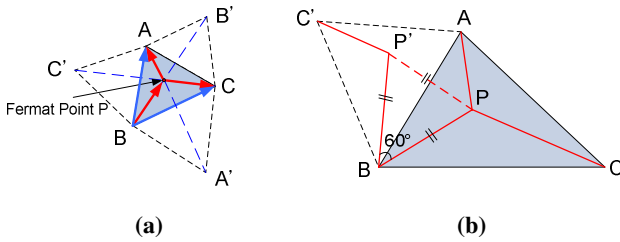
## 2 Background and Motivation

### 2.1 Definition of the Fermat Point

First of all, we need to explain the theorem of Fermat Point which is important to understand our scheme. Fermat point is the solution for the following question: *what is the point such that the sum of its distances from the vertices of a triangle is a minimum?* The definition of Fermat Point with a proof is following.

**Definition.** In any triangle  $\triangle ABC$ , we can draw three equilateral triangles  $\triangle A'BC$ ,  $\triangle B'CA$ , and  $\triangle C'AB$  at the edges of  $\triangle ABC$  as shown in Fig. 1(a). It is denoted by *Fermat Point*, an intersection point of three straight lines,  $\overline{AA'}$ ,  $\overline{BB'}$  and  $\overline{CC'}$ , each of which connects a vertex of the given triangle and a vertex of the opposite equilateral triangle.

**Theorem.** *Fermat Point is the point such that the sum of its distances from the vertices of a triangle is a minimum.*



**Fig. 1.** The definition of Fermat Point and its proof

**Proof.** In  $\triangle ABC$  in Fig. 1(b), select a point  $P$  and connect it with vertices  $A$ ,  $B$ , and  $C$ . Rotate  $\triangle ABP$   $60^\circ$  around  $B$  into position  $\triangle C'BP'$ . By construction,  $\triangle BPP'$  is equilateral,  $\overline{PA} = \overline{C'P'}$ , and  $\overline{PB} = \overline{P'B}$ . Thus, we have  $\overline{PA} + \overline{PB} + \overline{PC} = \overline{C'P'} + \overline{P'P} + \overline{PC}$ . As the image of  $A$  under the rotation, position of  $C'$  does not depend on  $P$ . Also,  $\overline{PA} + \overline{PB} + \overline{PC} \geq \overline{CC'}$  because the broken line  $\overline{C'P'P' + PC}$  is no shorter than the straight line  $\overline{CC'}$ . Therefore,  $\overline{PA} + \overline{PB} + \overline{PC}$  reaches its minimum if  $P$  lies on  $\overline{CC'}$ . For this  $P$ ,  $\triangle ABC'$  is also equilateral because  $\overline{AB} = \overline{C'B}$  and  $\angle ABC' = 60^\circ$ . With similar methods, we can draw other straight lines which connect vertices of the triangle with the opposite vertices of equilateral triangles. These straight lines cross at one point. From the definition, this point is Fermat Point. Thus, Fermat Point is the point such that the sum of its distances from the vertices of a triangle is a minimum.

The construction of Fermat Point fails if one of the internal angles of  $\triangle ABC$  is  $120^\circ$  or more. Because Fermat Point is drawn outside of the triangle. In this case, the vertex itself having the largest angle becomes an optimal point to reach each vertices of that triangle.

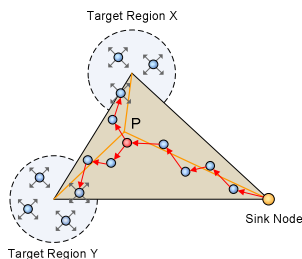


Fig. 2. Fermat Point applied in Sensor Networks with multiple target regions

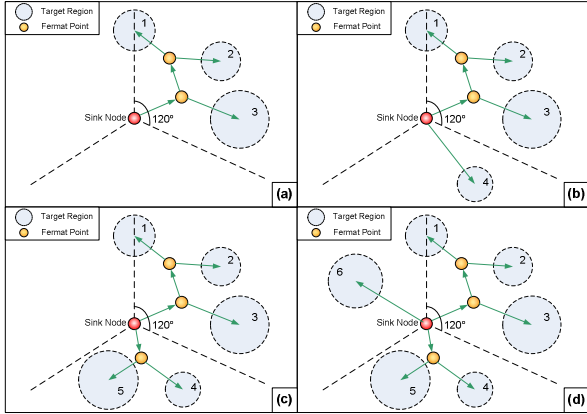
### 2.2 Applying Fermat Point Concept for Sensor Network Geocasting

We illustrate how to apply the theorem of Fermat Point in wireless sensor networks. Suppose that the two points A and B of the given triangle in previous Fig. 1(b) are the two central positions of the target regions X and Y, and then the point C is assumed to be the sink node. In this environment, the virtual triangle with three points is still formed as in Fig. 2. Therefore, we can calculate the Fermat point P, and a certain node which is closest to the Fermat Point can play a role as the junction point toward two target regions. From the sink node to this junction node, interest messages are delivered through the shared path, and then they are separated to each target region respectively. Thereby, we can optimize the interest forwarding process.

The above technique can be generalized to any number of increasing target regions as shown in Fig. 3. It may be possible to construct only one path curved severely like a circle when many target regions are placed in around the sink node. In this case, that path is extremely skewed and too long, so using such a path result in inefficiency. To solve this problem, we divide target regions into three different groups according to the angle made by them and the sink node. In Fig. 3(a), the target regions 1, 2, and 3 belong to the first group, since the angles made by the sink node and these regions do not exceed 120 degrees. In Fig. 3(b), and (c), the regions 4, 5 belong to the second group, since the angles made by the sink node and these regions are in between 120 degrees and 240 degrees. In the same way, region 6 belongs to the third group. The three shared paths are set up with target regions in each group respectively. Consequently, the interest messages initiated by sink node are sent along to each path.

## 3 Proposed Scheme: FERMA

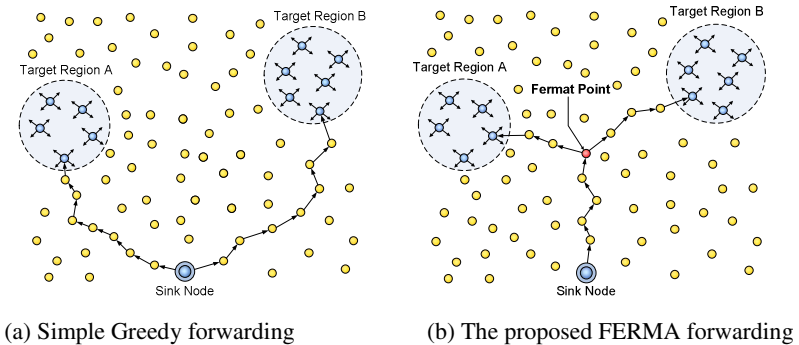
Our proposed scheme consists of two phases: interest forwarding phase and data forwarding phase. To disseminate interest messages toward multiple regions, a sink first creates a shared path based on the theorem of Fermat Point. Then, according to this path, interest messages are delivered to each target region. Any node receiving interest messages simultaneously sets up the corresponding gradient toward the previous sender for data forwarding in the next stage. More details will follow.



**Fig. 3.** Construction of the shared path in 360 degrees

### 3.1 Interest Message Forwarding Phase

For simplicity, let us consider only two target regions. Fig. 4 (a) illustrates a simple greedy forwarding approach towards the two different target regions. Since there is no optimization rule for multiple target regions in the pure greedy forwarding, the sink node is required to send an interest towards target region A and then, again sends the same interest message towards another target region B. When these interest packets reach in the designated target regions, local flooding is triggered, which means that all nodes within the target regions rebroadcast the receiving interests. Note that, as the number of target regions increases, the frequency of interest message transmissions by the sink also increases in this type of simple greedy approach.



**Fig. 4.** Comparison of Simple Greedy protocol and FERMA protocol

On the other hand, the proposed FERMA algorithm makes a virtual triangle with three vertices including the sink node and two central points of the target regions as mentioned in the previous section. From the definition of the Fermat Point theorem, that point becomes the optimal point that minimizes the sum of distances from the

vertices of a given triangle. It is clear that the shared path with the Fermat point is more efficient than separated multiple greedy forwarding (Compare Fig. 4(a) and (b) below).

After constructing shared path at the sink node, the interest packet is generated embedding three geographical coordinates with Fermat point and two central points of the target regions. The sink node sends this packet towards the Fermat point as the destination. Upon receiving the interest packet, each node selects the closest node to the destination as next hop among its neighbors. (For load balancing, when the node selects next hop, it can consider the amount of energy of neighbors as well as the distance to the destination. The residual energy of each neighbor can be informed by hello message) When a node could not find its neighbor node which is closer to the destined Fermat point, it becomes Fermat point by itself. Through this mechanism, the interest forwarding to Fermat point never fails because geographically closest node is always selected as the junction point. The next step is splitting of interests toward two target regions. Separated interest packets are forwarded to each target region respectively. Finally, when the interest reached any node inside the target region, local flooding is started. Local flooding means these flooded packets do not leak out of the target region. If any node located outside of the target region receives these locally flooded packets, it simply drops the packets.

Let us consider a generalization in more than two target regions. The basic idea is to chain consecutive Fermat points. Fig. 5 illustrates three target regions. Here, we calculate the first Fermat point 'F1' from the virtual triangle formed with the two target regions A, B and the sink. Next, the second Fermat point 'F2' is obtained with the first Fermat point F1, the target region C and the sink node. In this way, we can combine additional target regions into the previously constructed path. Ultimately, our scheme is scalable with the large number of target regions as mentioned in section 2.2.

After calculating the path, the sink node sends the interest towards the second Fermat point F2 with the *intermediate location list* including F2, C, F1, B, and A as seen in Fig. 5. The intermediate location list is filled with all locations that this packet goes through including the Fermat points and target regions. When this packet reaches F2, it is split into two packets, and then one is transmitted to the first Fermat point F1 with reduced intermediate location list including F1, B, A, and another is transmitted to the target region C. Arriving at F1, this packet is also separated into two packets for the target regions A and B.

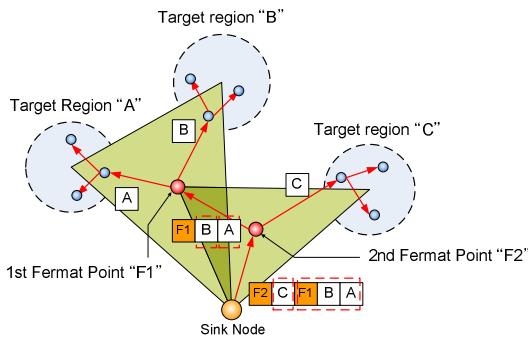


Fig. 5. Interest forwarding along the shared path

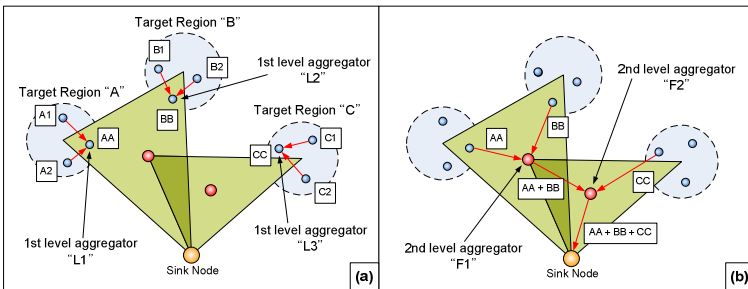
### 3.2 Data Packet Forwarding Phase

In the process of the interest packet forwarding, all nodes record the previous hop node which forwards that interest. It is called *gradient*, and used in several sensor network protocols, for example directed diffusion. If any node in the target region has data matched with the condition recorded in the interest packets, it sends these data to the sink node through the reverse path according to the gradient. Every intermediate node keeps records of the gradient, so packet can be delivery up to the sink node.

Some kind of data aggregation, for example the suppression of duplication, does not need any specific points to collect data packets. It is simply performed in any nodes using data cache. On the other hand, there are other aggregation functions as minima, maxima, average, etc. These functions are able to produce more exact and reliable results when a number of sensor nodes collaborate together. In this case, some intermediate nodes are required to collect data packets and to execute aggregation functions. Generally, in the cluster head or the intersection between multi-paths, the data aggregation is performed.

In our scheme, data aggregation is achieved as two levels. The first level aggregation is performed within each target region. In Fig. 6(a), the node L1, which is called to the *first level aggregator*, receives data packets A1 and A2 from the same region. After a while, L1 transmits the aggregated packet AA towards the sink node. This function is also executed in L2 and L3 in other target regions. Actually, these first level aggregators L1, L2 and L3 are the initial nodes starting the local flooding in the interest forwarding. These preliminary aggregated packets AA, BB and CC are aggregated ever further when they reach the Fermat points. Thus, F1 node aggregates the packets AA and BB, and also this aggregated packet is aggregated once more in *the second level aggregator* F2. These second level aggregator are exactly identical to the Fermat point nodes.

The data aggregation techniques adapted in our scheme does not require can reduce the total number of data packets without requiring any additional control packets. However, the latency of data delivery becomes a litter longer, because it needs to wait for the data packets during a certain time. The latency and the energy efficiency are trade-off. Therefore, in such a sensor network which requires more energy conservation than latency, our data aggregation techniques are definitely helpful.



(a) Data aggregation in each target region      (b) Data aggregation at Fermat point

**Fig. 6.** Two-level data aggregation



## 4 Performance Evaluation

In this section, we evaluate the performance of our scheme. Our primary performance metrics are the packet overhead and the energy consumption decreased due to path optimization toward multiple regions. Moreover, we are interested in how the performance of our scheme is affected by the variation of the number of target region.

In our simulation, the proposed scheme is compared to pure greedy forwarding and the flooding. We modify them, since they have no consideration of multiple target regions in interest delivery. In addition, they should be able to deliver the data packet from source node to sink node. Thus, we have augmented pure greedy forwarding with the multiple transmissions of interest packet toward each target region, local flooding within the target regions and data delivery through the gradient setup. In the flooding used in our simulation, the interest packet contains the location coordinates of all target regions. Thereby, the interest packet can be forwarded just once for multiple target regions. However, this method is still less efficient. The data forwarding is achieved through the flooding likewise interest forwarding.

### 4.1 Simulation Environment

We performed a simulation using *ns-2*. In our simulation model, 100 nodes are randomly placed in  $200m \times 200m$  square area. The transmission range of each node is  $40m$ . We consider the immobile sensor network, so every sensor node is static. We locate one sink node at coordinate  $(0, 0)$  for convenience. The number of target regions varies from 1 to 5, and the target region is circularly shaped with the radius of  $25m$ . The interest packet with fixed size payload of 36 bytes is periodically generated every 5 seconds from the sink node. In order to respond to the interests, the source node sends the data packet with fixed size payload of 64 bytes each. Total simulation time is 500 seconds and we repeat each scenario ten times with different randomly deployed nodes. We choose the following two metrics to analyze the performance of our scheme:

**Average Packet Overhead.** We measure the number of packets divided by the total number of nodes and the total number of queries generated from the sink node. The total number of queries is obtained by the simulation time divided by the interval of query generation. This metric reflects the overhead of packet transmissions by each node to deliver a query. We also examine the packet overhead on the aspect to packet size. It means that the total size of transmitted packets divided by the total number of nodes and the total number of queries generated from sink node.

**Average Consumed Energy.** We measure the total energy consumption of all nodes divided by the total number of nodes and the total number of queries in the same manner as the average packet overhead.

### 4.2 Simulation Results

The first evaluation is about the average packet overhead as a function of the number of target region. Fig. 8(a) shows the average interest packet overhead and Fig. 8(b) shows the average data packet overhead. In Fig. 8(a), when the interest is disseminated through flooding, we observe that every node transmits one interest packet per a query all the time. It means that every node has to attend to the interest

dissemination. Thus, flooding is the most inefficient method among three protocols that we examine. Our FERMA generates fewer interest packets than the greedy forwarding as much as maximum 29%. The difference of amount of interest packets is getting larger as the number of target region increases. It means that FERMA improve the performance much more as the number of source nodes increases. Such an improvement is achieved as our scheme uses the optimal shared path for multiple target regions. In addition, Fig. 8(b) shows FERMA forwards data packets more efficiently than the other two protocols. The two-level data aggregation enables the data packets from different source nodes to be collected and merged. Thereby, our scheme reduces a lot of the network traffic. Especially in the best case, our scheme produces fewer packets up to 1/15 of flooding, and 1/3 of greedy forwarding.

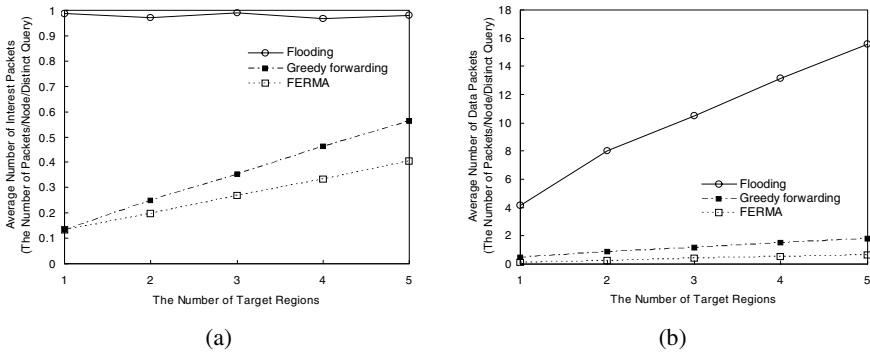


Fig. 8. Interest Packet Overhead and Data Packet Overhead

Fig. 9 shows the result of packet overhead on the aspect of the size of total packets including interest, data and hello messages. As seen in this figure, FERMA is more efficient than greedy forwarding as much as maximum 57%. Fig 10 represents the average consumed energy in three protocols. This figure shows that our scheme make the network nodes consume energy resources more efficiently than any others. Consequently, we can extend the life time of the network with the proposed scheme.

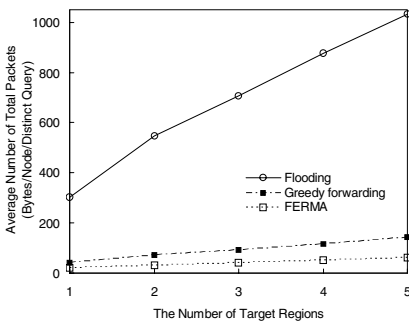


Fig. 9. Total Packet Overhead

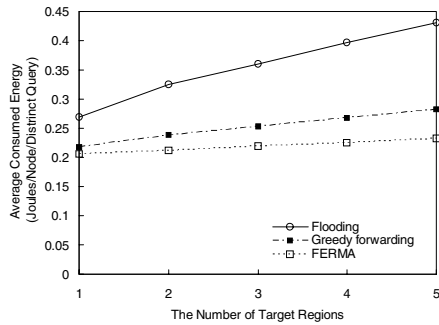


Fig. 10. Energy Consumption

## 5 Conclusion

In this paper, we focus on the efficient interest forwarding in the environment of multiple target regions. We propose a noble scheme which sends an interest message from the sink node at once instead of multiple packet transmissions towards different target regions respectively. The proposed FERMA makes a suitable shared path among multiple target regions using the theorem of Fermat Point. In addition, we examine the two-level data aggregation scheme to reduce more data packet overhead. Consequently, our scheme optimizes the delivery of interest messages and replied data. We prove such an improvement in terms of packet overhead and energy consumption through simulation studies. FERMA is useful in many applications and scenarios to desire interest dissemination frequently towards multiple target regions. Ongoing work includes avoiding the obstacle and exploring other situation in which performance degradation may occur.

## References

1. I.F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "Wireless Sensor Networks : A Survey", *Computer Networks*, vol.38, 2002, pp. 392-422.
2. J. N. Al-Karaki and A. E. Kamal, "Routing Techniques in Wireless Sensor Networks: A Survey", *IEEE Wireless Communications*, vol. 11(6), 2004, pp 6-28.
3. C. Intanagonwiwat, R. Govindan, D. Estrin and J. Heidemann, "Directed Diffusion for Wireless Sensor Networking", *IEEE/ACM Transactions on Networking*, vol. 11(1), 2003, pp. 2-16.
4. Y. Yu, R. Govindan and D. Estrin, "Geographical and Energy Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks", UCLA Comp. Sci. Dept. tech. rep., UCLA-CSD TR-010023, 2001.
5. *The Fermat Point and Generalizations*. [Online]. Available: [http://www.cut-the-knot.org/Generalization/fermat\\_point.shtml](http://www.cut-the-knot.org/Generalization/fermat_point.shtml)

# Power-Aware Position Vector Routing for Wireless Sensor Networks\*

Sangsoo Lee<sup>1</sup>, Daeyoung Kim<sup>1</sup>, Sungjin Ahn<sup>1</sup>, and Noseong Park<sup>2</sup>

<sup>1</sup> Real-time and Embedded Systems Laboratory,  
Information and Communications University (ICU),  
119 Munjiro, Yuseong-Gu, Daejeon, Korea, Postal Code: 305-714,  
Phone: +82-42-866-6812, Fax: +82-42-866-6810  
{sslee, kimd, sungjin}@icu.ac.kr

<sup>2</sup> Electronics and Telecommunications Research Institute (ETRI),  
161 Gajeong-Dong, Yuseong-Gu, Daejeon, Korea, Postal Code: 305-700  
Phone: +82-42-869-1735, Fax: +82-42-869-1711  
behack@etri.re.kr

**Abstract.** We propose PPVR, a power-aware position vector routing protocol for wireless sensor networks. PPVR is an energy efficient geographical routing protocol that improves existing geographic routing protocols in two ways. First, in the forwarding phase, PPVR reduces the transmission energy by passing packets through a relay node called a power-aware node. Second, PPVR provides an efficient void node avoidance scheme that mitigates the cases forwarding packets to a wrong path so that higher route discovery success rate is achieved. The simulation results shows that the PPVR outperforms existing geographic routing protocols including GPSR and GEAR, in terms of the energy consumption and the successful delivery rate.

## 1 Introduction

Wireless Sensor Network (WSN) is a network consisting of a number of tiny sensor nodes deployed in a region of interest, where each node is capable of processing and wireless communication. Because individual sensor node is small in size and memory, equipped sensors, the processing power and battery capability are limited, routing protocol developers must take care of all resource restrictions in order to provide the intended service. Of the restrictions, reducing energy consumption as much as possible is especially the most critical designing issue in developing a sensor network routing protocol.

As the location based applications [7, 10] in this field become more and more important and numerous methods to obtain the location information are developed in hardware (e.g., GPS) and software (e.g., distributed localization algorithms) ways [1, 8], routing protocols utilizing the geographical information are becoming popular [2, 3, 9]. The major advantage of these geographic routing protocols including GPSR [2] and GEAR [3] is that it minimizes the overhead of maintaining routing information

---

\* This research has been partially supported by ITRC project of Korea Ministry of Information and Communication.

(mostly, routing tables) since it selects the next node to forward packets only depending on the location of the neighbor nodes. However, this characteristic of selecting the next node by local optimal view causes a critical problem called *void area problem* that is occurred when a routing path reaches to a node which has only one neighbor that has just sent a packet to it. This problem has to be well dealt with to provide high success rate of packet delivery.

In this paper, we present a power-aware position vector routing protocol (PPVR) which consists of the energy efficient packet forwarding mechanism and the void node avoidance scheme. In the forwarding phase, PPVR uses power-aware position vector routing that forwards packets through a *power-aware node* that makes the total transmission energy be reduced even if it increases the hop count to the destination. In addition, to avoid the cases for a packet to reach a void area, the PPVR utilizes *gate nodes* that have the information of void nodes so that it selects a path that does not lead to a void node area before reaching there. As a result, PPVR delivers control and data packets highly energy efficiently at a high success rate.

The rest of this paper is organized as follows. In the next section, power-aware position vector routing is presented, and in Section 3 the void node avoidance scheme is discussed. The performance evaluation results are shown in Section 4. Finally, Section 5 concludes this work.

## 2 Power-Aware Position Vector Routing

The position vector routing (PVR) is based on a greedy forwarding using the geographical information. Unlike the existing greedy forwarding algorithms used in GPSR and GEAR where a node selects the next node as a node that has the minimum distance to the destination node among its neighbors, position vector routing protocol uses two items of information to select the next node, the distance and the angle to the destination. The procedure determining the next node is as follows. First, a source node  $O$  calculates the following function  $f(N_i, D)$  for each neighbor node  $N_i$  regarding to the location of the destination node  $D$ .

$$f(N_i, D) = |N_i| |D| \cos(\text{ANGLE}(N_i, D))$$

Here,  $|X|$  denotes the distance from the source node  $O$  to a node  $X$  and  $\text{ANGLE}(X, Y)$  denotes the angle  $\angle XOY$ . Then, the node which has the maximum value for the function  $f(N_i, D)$  is selected as the next node to forward packets to the destination. Fig. 1 illustrates this procedure. By the definition of the function, as the distance is shorter and the angle is narrow, a node is likely to be selected as the next node.

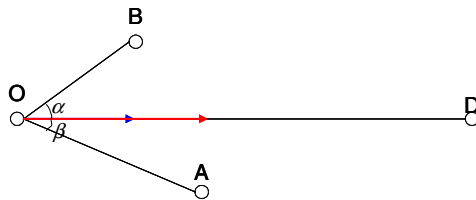
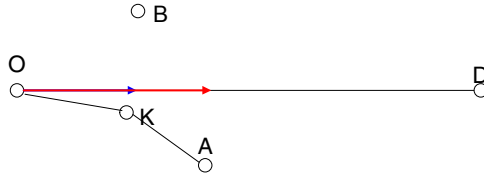


Fig. 1. Next node selection of PVR



**Fig. 2.** Power-aware node selection of PPVR

The position vector routing introduced above however does not consider the energy efficiency sufficiently when it determines the next node. Power-aware position vector routing protocol (PPVR) improves the PVR with consideration of energy efficiency of sensor networks.

The basic concept of the PPVR is that passing a packet by a neighbor node, called a *relay node* or a *power-aware node*, to a next node is more efficient in terms of transmission energy consumption even if the next node can be reached directly and passing by the relay node increases the hop count to the destination. Fig. 2 shows the case where passing a packet by a relay node *K* is more efficient than transmitting it directly to the next node *A*. Now we present the procedure that a node finds the power-aware node for a next node in more detail.

When nodes are deployed, they first start to acquire the location information of its neighbor nodes. After this procedure, each node then attempts to acquire the power-aware nodes for all neighbors. The power-aware node selection function, which calculates the expected energy consumption, depends on what power consumption model is used. The most commonly used power consumption models are the well-known Friss free space model and two ray ground model. In these models, the transmission energy is proportional to the distance between the transmitter and the receiver. Exactly speaking, the consumed power  $P(d) \propto 1/d^n$ , and  $n=2$  and  $n=4$  are used for Friss free space and two ray ground model, respectively [4].

According to the power consumption model, a node *O* given a next node *A* by the function  $f(N_i, D)$  calculates the following power consumption function  $f_{power}$  for all its neighbors  $N_i$ . V. Rodoplu et al [6] presented an assumption which is helpful to reduce the power consumption

$$f_{power}(N_i) = d(O, N_i)^n + d(N_i, A)^n + 2recv(n)$$

This function denotes the expected power consumption when a node *O* transmits a packet through a neighbor node  $N_i$  to a next node *A*, and  $d(X, Y)$  denotes the distance between nodes *X* and *Y*,  $recv(n)$  denotes the energy consumption at the receiver. There are twice receptions at node  $N_i$  and *A*. And the *n* becomes 2 or 4 depending on the underlying power consumption model. Then, if expected power consumption through a power-aware node  $N_k$  is less than that of directly sending to the next node *A*, the node *O* selects the neighbor node  $N_k$  as the power-aware node relaying packets to the next node *A*. If there is no node satisfying above condition, it directly sends packets to the next node *A*. The energy consumption of directly sending to *A* is:

$$d(O, A)^n + recv(n).$$

Therefore, the neighbor node satisfying the following condition and has the minimum power consumption becomes the power-aware node for the next node *A*.

$$d(O, N_i)^n + d(N_i, A)^n + \text{recv}(n) < d(O, A)^n$$

Fig. 3 shows this power-aware node selection procedure. Node  $O$  is the source and  $D$  is the destination, node  $R$ ,  $A$  and  $N_i$ s are neighbors of node  $O$ , and dashed circle shows the communication range of node  $O$ . Node  $A$  is selected as the next node for the destination node  $D$  by node  $O$  through the position vector routing presented previously and node  $R$  indicates the power-aware node between nodes  $O$  and  $A$ .

Figure 4 depicts the region where the power-aware node exists between  $O$  and  $A$ . All the nodes in the power-aware region circle which includes node  $R$ , satisfies the equation  $d(O, R)^2 + d(R, A)^2 + \text{recv}(n) < d(O, A)^2$ , when  $n=2$ .

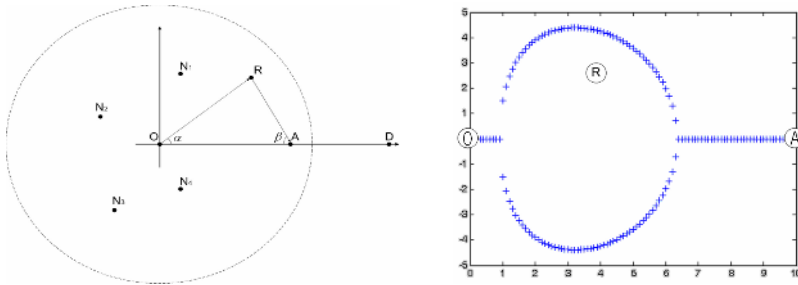


Fig. 3. Power-aware node selection Fig. 4. Power-aware region between node  $O$  to node  $A$

### 3 Void Node Avoidance

#### 3.1 Void Node Area Problem

Geographical routing protocols select the next node using its local information. That is, it finds the solution that is optimal only in local. For example, GPSR selects the next node as the one which has the minimum distance to the destination, among its neighbors. However, even if it is locally optimal, it cannot guarantee the optimality in global. Thus, a routing path proceeded through the local optimal strategy may lead to extensively inefficient path or even route discovery failure, depending on the density and the topology of the network. *Void node area* is an area where a node does not have any neighbor to send a packet except the one from which it has just received the packet. When a route packet reaches to this area, it takes many steps to escape the area and then to find proper path toward the destination.

Fig. 5 shows an example of the void node area. Source node  $S$  sends a route discovery request to the destination node  $D$  using a geographical routing. When the packet reaches to the node  $a_2$ , it selects  $b_1$  as the next node since it is the closest node to the destination among  $a_2$ 's neighbors (here,  $a_3$ ) and thus becomes the local optimal solution. However, it leads to the void node area since  $b_2$  is a void node. Thus, it is better for  $a_2$  to select  $a_3$  instead of  $b_1$  as the next node avoiding reaching to the void node.

GPSR and GEAR introduced solutions for this problem that escapes the void node area after reaching to the area. However, we here present an improved solution which selects the correct path avoiding entering to the void node area.

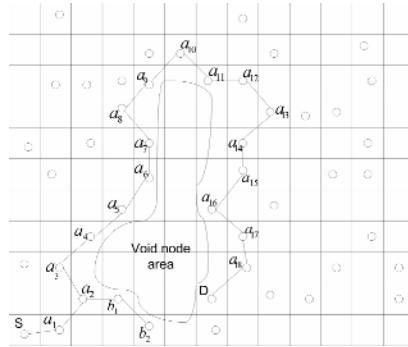


Fig. 5. Void node area problem

### 3.2 Void Node Avoidance Scheme

*Void node* is a node that has only one neighbor that has just sent a packet to it. Therefore, when a routing packet reaches to a void node, it cannot progress anymore. *Gate node* is a node that is the closest to the *void node* and having more than three neighbors. A set of connected nodes is said to be a *node chain*, if and only if a node  $a_k$  in the chain has only two neighbors  $a_{k-1}$  and  $a_{k+1}$ , except the two nodes at the ends of the chain, which are a gate node and a void node. A node chain is denoted as  $a_1 \leftrightarrow a_2 \leftrightarrow \dots \leftrightarrow a_k \leftrightarrow \dots \leftrightarrow a_n$ . For example, in Fig. 5,  $b_2$  is a void node and  $a_2$  is a gate node. Thus, the connected nodes,  $a_2 \leftrightarrow b_1 \leftrightarrow b_2$ , form a node chain. Now we present the detail void node avoidance algorithm.

After the acquisition of neighbor locations, a node can know whether it is a void node or not. If a node is turned out to be a *void node*, it sends a void region notification message to a *gate node* along its node chain. Then, a node receiving the notification message and has more than three neighbors becomes a gate node and there the notification ends. After void node acquisition and gate node selection procedure are completed, all gate nodes are aware of the void nodes and void area in their responsible territory. Thus, receiving a routing request packet, the gate node avoids guiding the packet toward the void node area by selecting the next node as the one that is not in any node chain. However, if the destination is in the middle of the node chain, this makes a problem.

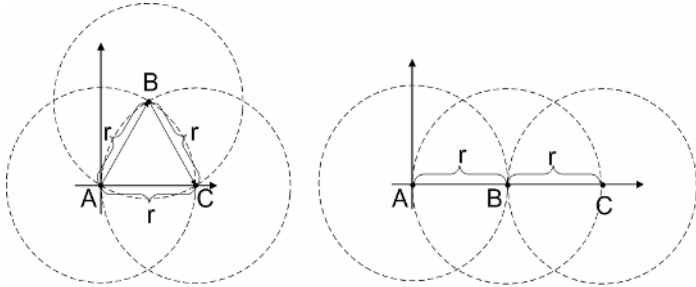
So, a gate node has to keep the information of the nodes in the chain. In some worse cases where the density of the network is low, the length of the node chain may become larger and this means that the amount of information a gate node should keep becomes larger as well. PPVR presents a mechanism that only keeps the information at most a half of the total number of nodes in a node chain by the following theorem.

**Theorem 1:** Given a node chain,  $a_1 \leftrightarrow a_2 \leftrightarrow \dots \leftrightarrow a_{k-1} \leftrightarrow a_k$ , the gate node  $a_1$  only requires  $\text{floor}(k/2)$  number of nodes information to discover all the nodes in the chain, where  $k$  is the number of nodes in the chain.

**Proof:** By the definition of the node chain, a node  $a_i$  has only two neighbors  $a_{i-1}$  and  $a_{i+1}$ , and the two neighbor nodes cannot communicate with one another. Thus, as shown in Fig. 6, given the locations of two nodes A and C, and that they are out of



communication range  $r$  each other, but share a node, it can be inferred that there is only one node (here  $B$ ) between the intersection range of  $A$  and  $C$ . This means that it does not require knowing the location information of node  $B$  when we know the location of  $A$  and  $C$ .



**Fig. 6.** Reducing location information to  $\text{floor}(k/2)$

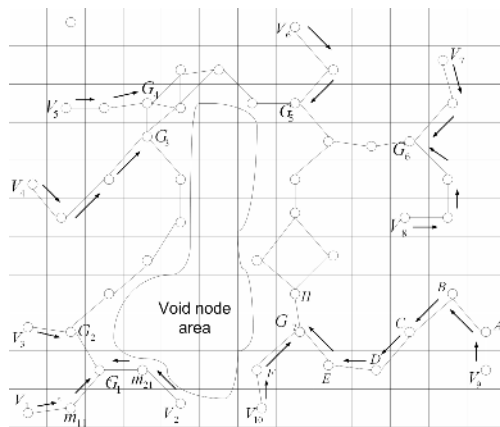
Now, we compare how PPVR works with the void node avoidance scheme to without one.

**Without void node avoidance scheme**

In figure 7, assuming that  $V_1$  is the source and  $C$  is the destination.  $V_1$  chooses  $m_{11}$  as next node by position vector routing, then  $m_{11}$  selects  $G_1$ ,  $G_1$  select  $m_{21}$  as the next node by the local optimal strategy and then  $m_{21}$  according to position vector routing, and it finally reaches to  $V_2$  the void node. This leads to route discovery failure if proper void area avoidance mechanism is not provided even though there are alternative path going to  $G_2$  at  $G_1$ .

**With void node avoidance scheme**

If the void node avoidance scheme is supported, when it reaches to  $G_1$ , it selects  $G_2$  as the next node since  $G_1$  is aware that going through  $m_{21}$  leads to a void node and that



**Fig. 7.** PPVR with void node area avoidance scheme

the destination is not in the node chain. Then, following the same procedure, it finally reaches to  $G_7$  by the power-aware position vector routing. At  $G_7$ , it selects  $E$  as the next node even if it is a node chain consequently leading to a void node, since  $G_7$  is keeping the location information of  $V_9$ ,  $B$ , and  $D$ , and from this information it can infer that the destination is in the middle of  $D$  and  $B$  as explained in above theorem.

## 4 Performance Evaluation

### 4.1 Simulation Environment

The MICA2 [5] uses CSMA/CA MAC and 433 MHz RF chip that gives the maximum output power of 10 dBm with the sensitivity of  $-109$  dBm. It consumes 5.3 mA and 26.7 mA at its minimum and maximum signal strengths, respectively, and 7.4 mA in receiving mode. It can send at most 53 packets in a second with the packet size of 42 bytes. We assume that the antenna is omni-directional with 1 dB gain and located 10 cm above the ground. The maximum transmission range becomes 10 m by the specification. Two-ray model [4] is used for power consumption estimation. In a 50 m $\times$ 50 m sensor field, 200 sensor nodes are randomly deployed, for the evaluation of the local minimum problem. 10m $\times$ 10m void node area is deployed randomly in the sensor field.

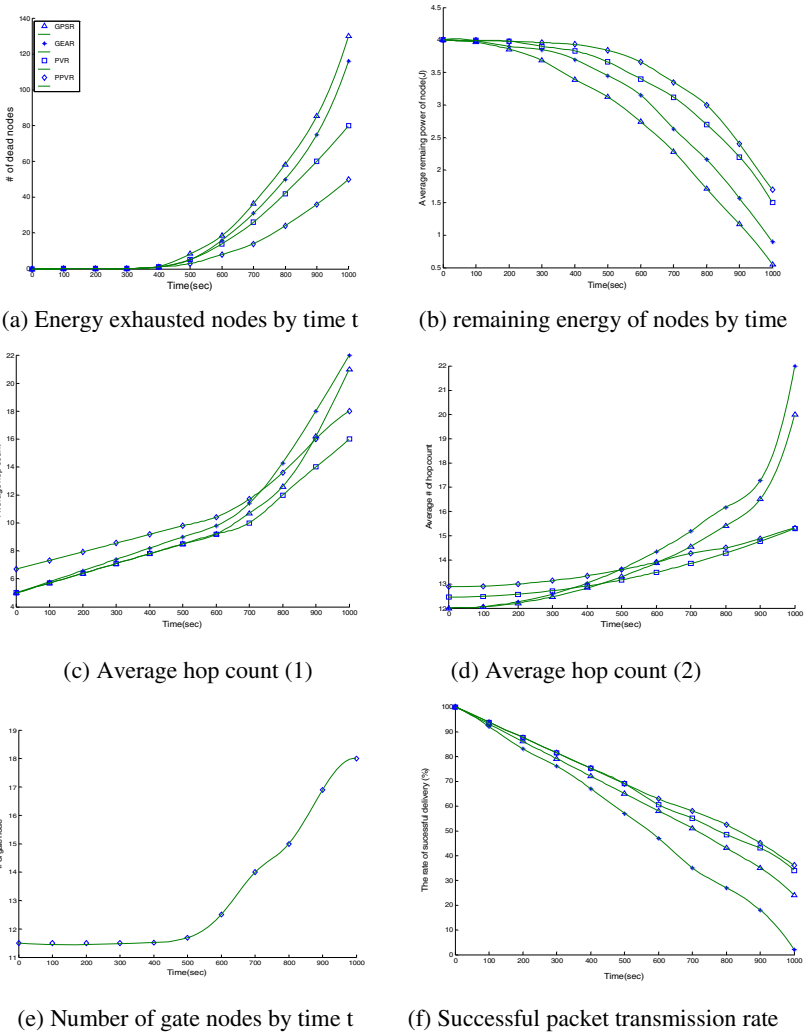
For the simplicity of simulation, each sensor node has 4J as its initial energy and sends data packets in a period ranging from 1 to 10 seconds (this is randomly chosen for each node). Considering the sensor node's restricted memory and poor communication conditions, the packet retransmission is not allowed, if the MAC layer fails to transmit any packet in a node. We compare the PPVR and PVR algorithms with the GEAR and GPSR algorithms.

### 4.2 Evaluation of Power Aware Node and Void Node Avoidance Scheme

We have evaluated our proposed idea in two different criteria. First, considering the energy consumption aspect, it shows how well our scheme saves energy as compared to the two existing schemes, GPSR and GEAR. Second, considering the ability avoiding the local minimum problem, it proves that how short the hop count would be and how the proposed avoidance scheme is effective for sensor networks.

PVR works basically the similar way to the GPSR, except for avoidance of local minimum. However, PPVR additionally uses the relay nodes, it can reduce more energy in packet transmission resulting in more alive nodes. In Fig. 8(a), PPVR and PVR prove that they can save larger amount of energy than others can do. Thanks to the transmission power control, when it is applied to the sensor network, it shows much saving of energy as PVR does. Furthermore, when power-aware node scheme is applied, PPVR outperforms GPSR and GEAR, and PVR as well. This means that as the power aware nodes are added in the initial PVR route, it further saves energy. Here, GPSR is the worst case due to the lack of energy saving facility.

When PPVR and PVR are applied, the number of nodes exhausting energy is smaller than the others. It results in the longevity of the sensor network. Now, we examine the average remaining energy of the sensor network with respect to time  $t$ . Fig. 8(b) shows PPVR and PVR guarantee higher average remaining energy.



**Fig. 8.** Simulation results: 50m x 50m sensor field with 200 nodes deployed randomly

Both Fig. 8(a) and 8(b) explain that PPVR is superior to others in reduction of energy consumption. However, the superiority must pay back with additional number of hop counts.

In Fig. 8(d), we can see the drawback of PPVR. It proves that the average number of hop counts of PPVR is greater than the other schemes in earlier time period. But, over its last lifetime, PPVR shows smaller hop counts than the others due to the void node avoidance scheme.

For more intensive evaluation of void node avoidance scheme, we simulated all the schemes deploying a 10m X 10m void node area randomly. In Fig. 8(d), in the early stage of the simulation, the average hop counts of PPVR is larger than the others. It

results from the effect of the power aware node. As regards the PVR, we can easily tell that its average hop count (which doesn't include the power aware node) is slightly less than GEAR and GPSR. However, PPVR becomes better as the number of packet transmission increases, thanks to the void node avoidance scheme. Thus, for overcoming the local minimum, void node avoidance scheme is more effective than the learning based avoidance by GEAR and the right hand rule by GPSR. To accomplish this significant performance improvement, we use the gate node approach. Fig. 8(e) shows the increase of the number of gate nodes with respect to time  $t$ . It is necessary for PPVR to have void node avoidance scheme.

When the void node avoidance scheme (which is the avoidance of local minimum) is applied and the power aware node is included to further save energy, there are two notable improvements – firstly, the longevity of the sensor network (which was proved by the early simulation result) and secondly, the improvement in successful routing.

In Fig. 8(f), it explains that PPVR and PVR outperform others in terms of the number of successful packet delivery rate (thanks to the void node avoidance scheme and power aware node).

## 5 Conclusion

In the paper, we propose a power-aware geographical routing protocol PPVR for sensor networks (which minimizes the energy consumption and results in the longevity of sensor networks) and local minimum avoidance scheme (which leverages successful packet delivery rate and minimizes average hop count considerably). The simulation results prove that PPVR improves the energy efficiency and packet delivery success rate compared to the existing geographical routing protocols including GPSR and GEAR.

## References

1. J. Hightower and G. Bordello. *Location systems for ubiquitous computing*. In IEEE Comp., 2001.
2. B. Karp and H. T. Hung, *GPSR: Greedy Perimeter Stateless Routing for Wireless Networks*, In ACM MOBICOM, 2000.
3. Y. Yu, et al. *Geographical and Energy Aware Routing: a recursive data dissemination protocol for wireless sensor networks*. In UCLA CS Tech. Report, 2001.
4. J. D. Kraus and R. J. Marhefka, *Third edition: Antennas for all applications*, Mc Graw Hill.
5. Crossbow Technology Inc. <http://www.xbow.com>
6. V. Rodoplu and T.H. Meng, *Minimum Energy Mobile Wireless Networks, IEEE J. Sel. Areas in Comm. (JSAC)*, 1999.
7. C. Intanagoniwat, R. Govindan and D. Estrin, *Directed diffusion: A scalable and robust communication paradigm for sensor networks*, In ACM MOBICOM, 2000.
8. N. Bulusu, J. Heidemann, and D. Estrin, *GPS-Less Low Cost Outdoor Localization for Very Small Devices*, IEEE Personal Comm. Magazine, 2000.
9. Y.B. Ko and N.H. Vaidya, *Location aided routing (LAR) in mobile ad-hoc networks*, In ACM MOBICOM, 1998.
10. A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao. *Habitat monitoring: Application driver for wireless communications technology*. In ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean, 2001.

# Multicast Routing with Minimum Energy Cost and Minimum Delay in Wireless Sensor Networks\*

Zhao Li, Wei Zhang, Hengchang Liu, Baohua Zhao, and Yugui Qu

Department of Computer Science, University of Science and Technology of China,  
Laboratory of Computer Science Institute of Software Chinese Academy of Sciences,  
Hefei, Anhui 230027, China  
{leezhao, bho, hcliu}@ustc.edu, {bhzhao, ygqu}@ustc.edu.cn

**Abstract.** In this paper, we consider the multicast routing with minimum energy cost and minimum delay (MEMD) in wireless sensor networks under the access control scheme of Spatial Time Division Multiple Access (STDMA). We formulate and explore both MEMD multicast and ME multicast, and show that the latter is just the Maximum Leaf Spanning Tree (MLST) problem, which is NP-complete. A 2-approximation algorithm is proposed for the MLST problem through improving a known one. Based on this algorithm, an approximation algorithm is obtained for the MEMD multicast problem. To further improve the delay result, we provide another approximation algorithm for our main problem using a different approach. These algorithms are all near linear in the size of the network graph, and also are shown to have good performance by simulation results.

**Keywords:** Wireless Sensor Networks, STDMA, MEMD multicast, MLST.

## 1 Instruction

In typical applications of wireless sensor networks [1], battery-powered sensors are scattered throughout a monitored area, left unattended and expected to operate for a long time. Therefore, routing protocols must be optimized for ultra-low power operation in such networks [2]. Moreover, protocol design is always constrained by the following limitations of wireless sensors [3]: limited battery energy, limited transceiver resources, limited frequency resources, limited processing capability and limited storage space. For wireless sensor networks, the Time Division Multiple Access (TDMA) can be used as the access control scheme, in which the transmission capacity is divided into time slots, and each direct link is assigned a dedicated slot. A promising approach for increasing its efficiency is the STDMA scheme [4], which takes into account that sensors are usually spread out geographically, and hence sensors with a sufficient spatial separation can use the same time slot for transmission. We propose the latter as the access control scheme in our concern.

---

\* This paper is supported by the National Natural Science Foundation of China under Grant No. 60241004, the National Grand Fundamental Research 973 Program of China under Grant No. 2003CB314801, and the State Key Laboratory of Networking and Switching Technology.

In wireless sensor networks, multicast routing, which refers to the transmission of the same information to several destinations, is receiving an increasing interest due to its foundational applications, such as data dissemination and control information delivery. In this paper, we study the problem of multicast routing with minimum energy cost and minimum delay. Given a wireless sensor network and a multicast request, our concern is to find a multicast scheme such that the total energy cost and delay of the multicast is minimized. To our best survey, there are still no routing schemes proposed for the multicast considering these two objects together.

The contribution of our work is three-fold: Firstly, we formulate and explore the MEMD multicast problem and the ME multicast problem, and show the latter is just the MLST problem which is NP-complete. Secondly, we propose an improved 2-approximation algorithm for the MLST problem, based on a known one. Finally, we present two approximation algorithms for the MEMD multicast problem. These algorithms are all near linear in the size of the network graph and are shown to have good performance by both theoretical analysis and simulation results.

The rest of this paper is organized as follows: Section 2 posts out the related work about multicast routing and the MLST problem. We formulate and explore the MEMD multicast problem and the ME multicast problem in section 3. In Section 4, we present an approximation algorithm for the MLST problem and two other different ones for the MEMD multicast problem. Section 5 shows simulation results. We conclude this paper in Section 6.

## 2 Related Work

In recent years, many multicast routing schemes [3], [5-9] have been proposed for Ad Hoc Networks and wireless sensor networks, but none of them has considered minimum energy cost and minimum delay together. The directed diffusion scheme [8] and the two-tier data dissemination (TTDD) scheme [9] naturally support data multicasting, but they are not efficient. To improve the efficiency, the tree-based multicasting scheme can be used. In this scheme, the source and the destinations form a tree rooted at source, and the source pushes data to the destinations along the branches. In [6], Jia and Li et al discussed the energy efficient tree-based multicast problem in ad hoc wireless networks. They proved the problem is NP-hard and gave three heuristic algorithms. However, they didn't concern the multicast delay.

Given a connected undirected graph  $G$ , the MLST problem is to find in  $G$  a spanning tree with maximum number of leaves. Using reduction from the dominating set problem Garey and Johnson have shown in [10] that this problem is NP-complete, thereafter Galbiati et al established its MAX-SNP completeness [11]. Thus there exists some constant  $\varepsilon > 0$  such that there is no  $(1 + \varepsilon)$ -approximation algorithm for MLST unless  $P = NP$  [12], [13]. In [14], Lu and Ravi, using a novel approach based on a notion of leafy trees, obtained a 3-approximation algorithm which works in near linear time. Solia-Oba, in [15], presented a linear 2-approximation algorithm for MLST. This algorithm first constructs a leafy forest [14]  $F$  with certain expansion rules and then connects the trees of  $F$  and all vertices not in  $F$  to form a spanning tree of the graph  $G$ . To our best survey, this algorithm is the best one for this problem. The full

algorithm is shown in the Appendix. Base on this algorithm, we propose a more efficient one in Section 4.

### 3 Mathematical Programming Formulations

Energy cost minimizing and resource allocation in STDMA [16] both are challenging problems in multicast routing design. In this section, we define and formulate the MEMD multicast problem. To further study the multicast problems, we also consider the objectives respectively and explore the ME multicast problem.

#### 3.1 Assumptions

We consider a wireless sensor network  $G$  with nodes set  $V$ , where each node of  $V$  represents a stationary sensor node. STDMA is proposed as the access control scheme. Following the assumptions in [17], a direct communication link can be established between two nodes if their corresponding signal-to-noise ratio (SNR) is not lower than a certain threshold. There are several constraints and restrictions when assigning time slots. Firstly, a node can transmit or receive, but not both, in a time slot. Secondly, a node can receive data from at most one node at a time. Finally, a link is error-free only if the signal-to-interference ratio (SIR) is not lower than a threshold.

Without loss of generality, we make simplified assumptions as follows.

- All nodes broadcast multicast information at the same longest transmission radius  $R$ .
- A direct communication link between two nodes is established if their Euclidean distance is shorter than  $R$ .
- Two nodes can transmit in the same time slot without interference if the intersection of their neighbor sets is empty.
- The source node, which may be a cluster leader, a gateway or a region agent, knows the whole network topology.
- The energy cost of multicast routing can be measured only by their transmission energy since other energy cost such as receiving cost and idle cost can be simply assumed to be the same for all multicast routings.
- The multicast delay, which is the interval between the time the source nodes broadcasts the information and the time the last destination node receives it, can be measured by the total number of time slots.

Consequently, the multicast problem is to generate a spanning tree of  $G$ , rooted at the source node. In practice, the root generates the multicast tree and the assignment of time slots, and promulgates these schedules to all other nodes. Apparently, the time slot of a node must be behind the one of its parent node in the multicast tree. In the process of multicast routing, the root is the sponsor and all other branch nodes broadcast received multicast information at radius  $R$  in their own time slots.

#### 3.2 Formulation for MEMD Multicast

Under the assumptions in the section above, every branch node of the multicast tree has to broadcast the multicast information. So the multicast energy cost can be measured by

the total number of branch nodes. Further more, each branch node should be assigned one time slot under the constraints mentioned above, and hence the total number of time slots can be used to measure the multicast delay. Consequently, the MEMD multicast problem is to generate a spanning tree of the network graph with minimum branch nodes and minimum total number of time slots. However, these two objects have no direct relation and can not be obtained at the same time. To further explore the MEMD multicast problem, we formulate it in a combinatorial optimization model in this section.

Since the multicast is tree-based, we first present a set of formulations for multicast tree (MCT), which is a spanning tree of the network. We introduce the following variables.

$$Link((u, v)) = \begin{cases} 1 & \text{if } (u, v) \text{ is an in link to } u \text{ in the result graph} \\ -1 & \text{if } (u, v) \text{ is an out link to } u \text{ in the result graph} \\ 0 & \text{otherwise: } (u, v) \text{ is not a link in the result graph} \end{cases}$$

$$ConnectRoot(u) = \begin{cases} 1 & \text{if } u \text{ is connected with the root in the result graph} \\ 0 & \text{otherwise} \end{cases}$$

The formulations set for multicast tree is as follows.

$$[MCT]: G = (V, E, r)$$

$$|InLink(u)| = 1, \forall u \in V, u \neq r; \tag{1}$$

$$ConnectRoot(u) = 1, \forall u \in V; \tag{2}$$

$$Link(e) \in \{-1, 0, 1\}, \forall e \in E; \tag{3}$$

$$Link((u, v)) + Link((v, u)) = 0, \forall (u, v) \in E, u, v \in V; \tag{4}$$

$$Link((r, v)) = -1, \forall (r, v) \in E, v \in V; \tag{5}$$

$$InLink(u) = \{v \mid Link((u, v)) = 1, (u, v) \in E\}, \forall u \in V; \tag{6}$$

$$ConnectRoot(u) \in \{1, 0\}, \forall u \in V; \tag{7}$$

$$ConnectRoot(r) = 1; \tag{8}$$

$$ConnectRoot(u) = 1, \forall u \in V, \exists v \in V \text{ } ConnectRoot(v) = 1, Link((u, v)) = 1; \tag{9}$$

In formulations set MCT,  $G$  is a connected undirected graph, and both  $(u, v)$  and  $(v, u)$  express the same edge between nodes  $u$  and  $v$ .  $InLink(u)$  denotes the set of all the parent nodes of node  $u$ . Constraint (1) restricts that the result graph has no circle and constraint (2) requires all nodes of  $G$  to be connected with the root. Due to (1) and (2), the result graph must be a spanning tree of  $G$ . Constraint (9) presents the connection rule that a node is connected with the root only when its parent is connected with the root formerly. Other constraints define the variables and initialize them.

The formulation for MEMD multicast is shown in Formulation I.  $TS(u)$  denotes the time slot assigned to node  $u$ ,  $Zone(u)$  expresses the set of nodes that is in the transmission region of  $u$ ,  $BranchNodes$  denotes the set of all branch nodes, and other terms retain the same meaning in the above formulations.



**Formulation I:** MEMD Multicast  $G = (V, E, r)$

Object:  $\text{Min}(\alpha | \text{BranchNodes} | + \beta \text{Max}_{u \in \text{BranchNodes}}(TS(u))), \alpha, \beta \in [0, 1], \alpha + \beta = 1$  (10)

Subject to:

MCT ; (11)

$TS(v) > TS(u), \forall u, v \in V, \text{Link}((u, v)) = -1;$  (12)

$\text{Zone}(u) \cap \text{Zone}(v) = \emptyset, \forall u, v \in V, u \neq v, TS(u) = TS(v);$  (13)

$\text{BranchNodes} = \{u \mid \sum_{v:(u, v) \in E} \text{Link}((u, v)) \leq 0, u \in V\};$  (14)

$TS(u) \in \{1, 2, \dots, n\}, \forall u \in \text{BranchNodes};$  (15)

$TS(r) = 1;$  (16)

$\text{Zone}(u) = \{u\} \cup \{v \mid (u, v) \in E, v \in V\}, \forall u \in V;$  (17)

In Formulation I, the objective function (10) jointly minimizes the multicast energy cost and delay, where  $\text{Max}_{u \in \text{BranchNodes}}(TS(u))$  expresses the maximum time slot equaling to the total number of time slots in best schedules, while  $\alpha$  and  $\beta$  are application-specific coefficients. Constraints (11), the multicast tree formulations, ensure that the result graph of Formulation I is a spanning tree of the network. Constraints (12) and (13) express the constraints of assigning time slots mentioned in the section above. Constraint (12) restricts that the time slot of a node must be behind the one of its parent node and constraint (13) ensure that nodes assigned the same time slot do not cause transmission interference. Other constraints define the variables and initialize them.

From Formulation I, we notice the MEMD multicast is a complicated problem and that it needs exponential running time to get a complete solution. To further explore the MEMD multicast, we will formulate and study the ME multicast problem in the next section.

**3.3 Formulation for ME Multicast**

Actually, considering only one object of minimum energy cost, the multicast problem is to find a spanning tree with minimum branch nodes, which is just the MLST problem. Formulation II formulates this problem, and all symbols and constraints follow Formulation I.

**Formulation II:** ME Multicast  $G = (V, E, r)$

Object:  $\text{Min} | \text{BranchNodes} |$  (18)

Subject to:

MCT ; (19)

$\text{BranchNodes} = \{u \mid \sum_{v:(u, v) \in E} \text{Link}((u, v)) \leq 0, u \in V\};$  (20)

Actually, this problem is NP-complete as introduced in Section 2. Fortunately, there are already many efficient approximation algorithms. In the following section, we

propose an improved 2-approximation algorithm for this problem outperforming known ones.

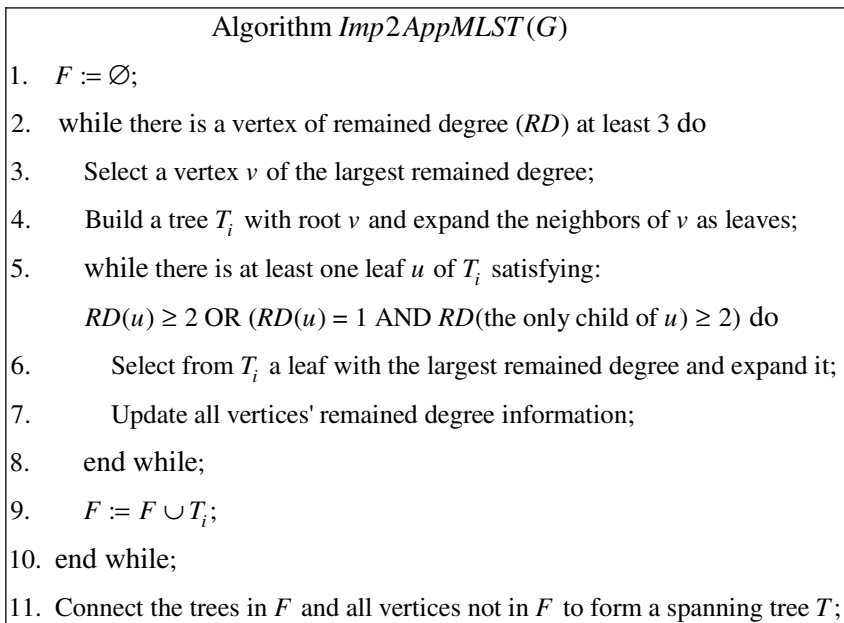
Intuitively, the MEMD multicast problem is NP-complete or NP-hard because the MLST problem is a special case of it (when  $\alpha=1$  and  $\beta=0$ ). Actually, the MEMD multicast problem is even too hard to use some novel approaches such as column generation for optimal or near-optimal solutions. In the following sections, we attempt to develop two approximation algorithms for the MEMD multicast problem in different approaches.

## 4 Approximation Algorithms

In this section, we first propose a improved 2-approximation algorithm for the MLST problem based on a known one, and then present two approximation algorithms for the MEMD multicast problem using different approaches.

### 4.1 An Improved 2-Approximation Algorithm for MLST

Based on the algorithm of [15], we propose an improved 2-approximation algorithm, *Imp2AppMLST(G)* (see Fig. 1), for the MLST problem. Like [15], this algorithm first constructs a leafy forest with certain expansion rules and then connects the trees of the forest and all vertices not in the forest to form a spanning tree of the original graph. We introduce *remained degree* to express the expansion rules of [15]. A vertex's *remained*



**Fig. 1.** An improved 2-approximation algorithm for MLST

*degree* is the number of its connections that point to unexpanded vertices. The expanding vertices are sorted more elaborately by their *remained degree* than [15], where there are only two priorities: priority 1 and priority 2 for expanding vertices. Moreover, the trees are constructed orderly by their *remained degree* too.

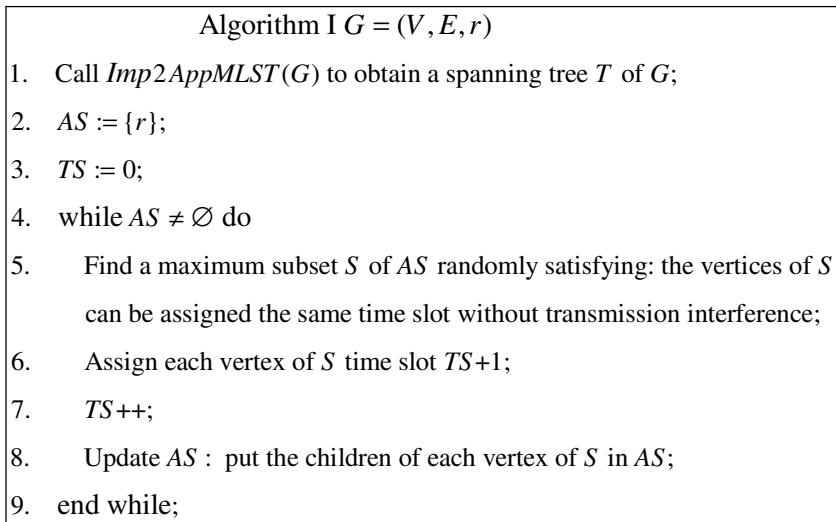
In line 6 of *Imp2AppMLST(G)*, two expanding vertices are sorted by their only child's *remained degree* if they have the same *remained degree* 1. The *while* of line 5 continues if there is at least one leaf of  $T_i$  satisfying that its *remained degree* is greater than 1 or its *remained degree* is 1 but the *remained degree* of its only child is greater than 1. Due to this constraint, the result forest is leafy. Lines 5 and 6 also accord with the expansion rules of [15].

This algorithm is more efficient than the original one because it sorts the expanding vertices more elaborately. The tree generated by this algorithm is at least a 2-approximation MLST of the graph  $G$  since this algorithm is a special case of the original one. The running time  $\Theta(n^2 + m)$  required by this algorithm is almost linear in the size of  $G$  (nodes number and edges number), where  $n$  is the number of nodes and  $m$  is the number of edges.

## 4.2 Algorithm I for MEMD Multicast

In this section, we present Algorithm I (see Fig. 2) for the MEMD multicast problem. This algorithm first calls the algorithm *Imp2AppMLST(G)* to obtain a 2-approximation MLST of the network graph and then assigns time slots to all branch nodes based on the tree.

Algorithm I assigns time slots to all branch nodes round by round, spreading from the root.  $AS$  denotes the set of vertices that can be assigned time slots in a round and



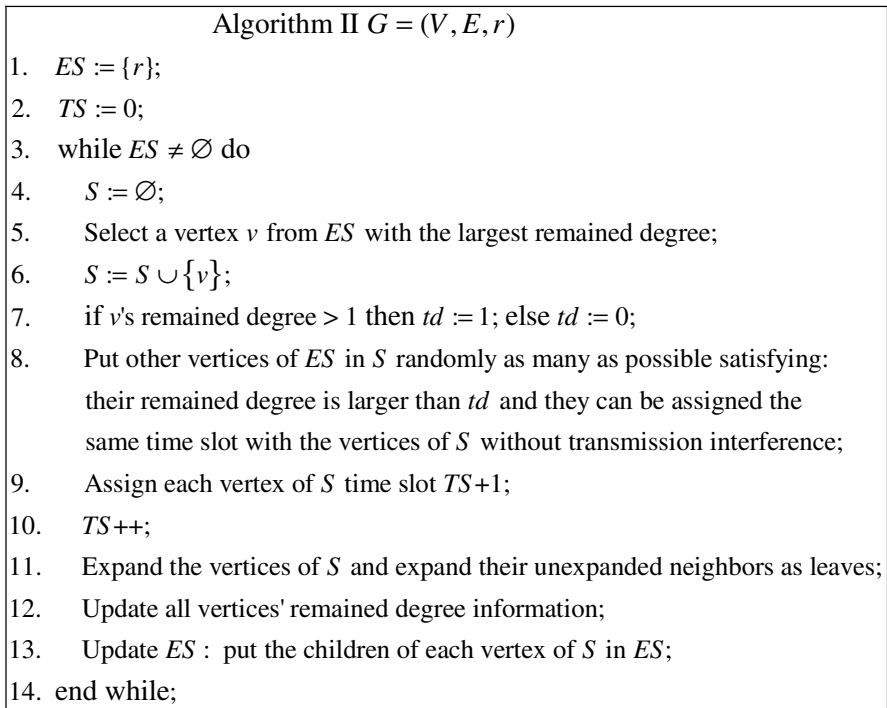
**Fig. 2.** Algorithm I for MEMD Multicast

should be updated round by round, while  $S$  denotes the vertices that can be assigned the same time slot without interference.  $TS$  represents the time slot. Actually, the distance between two nodes can be used to judge whether they will incur transmission interference. For example, two nodes can transmit in the same time slot without interference if their distance is longer than double the largest transmission radius. Also, a more elaborate approach like *Zone()* in Formulation I can be used to avoid transmission interference.

Due to the algorithm *Imp2AppMLST(G)*, Algorithm I can greatly minimize the energy cost. However, the delay may not be minimized enough because MLST may not be very symmetrical and hence can not reuse time slots efficiently. The running time of the process of assigning time slots is  $\Theta(n^2)$ , so Algorithm I is near linear with a running time of  $\Theta(2n^2 + m)$ .

### 4.3 Algorithm II for MEMD Multicast

According to the analysis in the section above, as Algorithm I considers the energy cost and the delay respectively, the latter object may not be optimized enough. To improve delay result, we offer a new approach in Algorithm II (see Fig. 3) for MEMD multicast, which assigns time slots in the generation process of the approximate MLST of the network graph.



**Fig. 3.** Algorithm II for MEMD multicast

In Algorithm II,  $ES$  denotes the set of expanding vertices. Other symbols remain the same meanings in Algorithm I. Line 5 ensures that the current tree can be expanded even if all leaves' *remained degree* are only 1, so Algorithm II can expand all vertices and get a spanning tree of  $G$ . Lines 7 and 8 are intent on generating a subset  $S$  of  $ES$  with vertices as many as possible satisfying that all the vertices of  $S$  can be assigned the same time slot without transmission interference.

Unfortunately, in worst cases when the sensors are sparse, the spanning tree generated by Algorithm II is not a 2-approximation MLST of  $G$ , because this algorithm can not guarantee that the result tree is leafy. However, in general cases, especially in dense wireless sensor networks, the total number of branch nodes of the tree should be close to the result of Algorithm I. Moreover, this algorithm can obtain a better result on delay because it considers this object more elaborately. Simulation results in the next section confirm this analysis. This algorithm is no more complex than Algorithm I even in worst cases. The running time required by this algorithm is  $\Theta(n^2 + m)$ .

### 5 Simulation Results

We propose a wireless sensor network with 200 sensors distributed randomly in a variable square place. The longest transmission radius is 120m. A direct communication link between two nodes is established if their distance is shorter than 120m. The link obstacle rate is set to one percent. Moreover, two nodes can transmit in the same time slot without interference if their distance is longer than  $2*120m$ , with a more precise threshold  $\sqrt{3} * 120m$  for the root's neighbors.

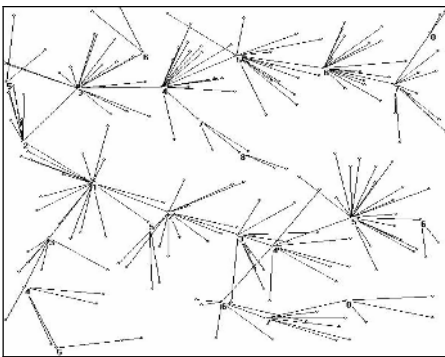


Fig. 4. A sample solution obtained by Algorithm II

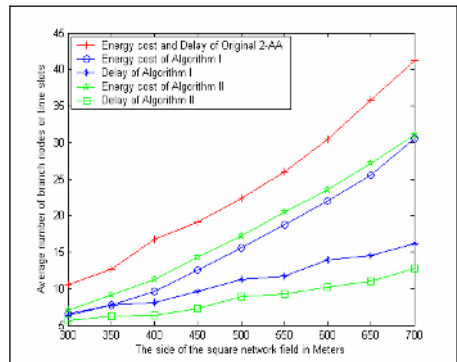


Fig. 5. Simulation results (Original 2-AA denotes the original 2-approximation algorithm for MLST)

A solution example obtained by Algorithm II is shown in Fig. 4. The total number of branch nodes is 24 and only 8 time slots are required. From this example, we have a vivid impression that the energy cost is low and the time slots are reused efficient. Next, the average results will be shown for the algorithms in the sections above.

By the way, if TDMA is proposed as the access control scheme, each branch node of the multicast tree is assigned a dedicated time slot, and hence both minimum energy cost and minimum delay can be measured by the total number of branch nodes. So the MEMD multicast based on TDMA is also the MLST problem. We implement the original 2-approximation algorithm for the MLST problem here as a solution for this multicast problem. We run the original 2-approximation algorithm, Algorithm I and Algorithm II for networks in different sizes, each of which consists of 1000 random topologies. The average results are shown in Fig. 5.

From the simulation results, we first notice that Algorithm I and Algorithm II work remarkably better than the original 2-approximation algorithm on both energy cost and delay results. Secondly, STDMA outperforms TDMA remarkably, especially when the sensors density is low. Thirdly, these results affirm that our 2-approximation algorithm for MLST, used in Algorithm I, is more efficient than the original one. Moreover, Algorithm I outperforms Algorithm II on energy cost while Algorithm II outperforms Algorithm I on delay. Finally, as mentioned in Section 4.3, the tree obtained by Algorithm II is not a 2-approximate MLST in worst cases, however, the average result is close to Algorithm I's and remarkably better than the result of the original 2-approximation algorithm. These results affirm our analysis in the sections above.

## 6 Conclusion and Future Work

In this paper, we have studied the problem of multicast routing with minimum energy cost and minimum delay in wireless sensor networks under STDMA. Firstly, we formulate and explore both MEMD multicast and ME multicast, and show that the latter is just the MLST problem which is NP-complete. Then a more efficient 2-approximation algorithm is proposed for the MLST problem through improving a known one. Based on this algorithm, we obtain a near linear approximation algorithm for the MEMD multicast problem. To further improve the delay result, we also propose another near linear approximation algorithm with a different method for our main problem. Simulation results show that both algorithms have good performance.

In the future, we plan to further study the resource allocation on multicast tree under STDMA and develop a distributed implementation for our multicast schemes. We also plan to consider the multicast problems in mobile wireless sensor networks and Ad Hoc networks.

## References

- [1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: A Survey," *Computer Networks*, vol. 38, no. 4, March 2002.
- [2] J. L. Hill and D. E. Culler, "Mica: a wireless platform for deeply embedded networks", *IEEE Micro*, vol. 22, no. 6, pp 12 – 24, 2002.
- [3] Xiaoxing Guo, "Broadcasting for Network Lifetime Maximization in Wireless Sensor Networks," 2004 First Annual IEEE Communications Society Conference, pp 352 – 358, 2004.

- [4] R. Nelson and L. Kleinrock, "Spatial-TDMA: A collision-free multihop channel access control", IEEE Transactions on Communications, vol. 33, pp 934 – 944, 1985.
- [5] Di.Prieto.R, Mancini.L, "LKHW: a directed diffusion-based secure multicast scheme for wireless sensor networks," Proceedings of the 2003 International Conference on Parallel Processing Workshops (ICPPW'03), pp 397 – 406, 2003.
- [6] Xiaohua Jia, Deying Li, "Multicast routing with minimum energy cost in ad hoc wireless networks," GLOBECOM '04, IEEE vol. 5, pp 2897 – 2901, 2004.
- [7] Peng-Jun Wan, Calinescu, G, "Minimum-power multicast routing in static ad hoc wireless networks," IEEE/ACM Transactions on Networking, vol. 12, Issue 3, pp 507 – 514, 2004
- [8] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication," MOBICOM '00, August 2000.
- [9] F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang, "A Two-Tier Data Dissemination Model for Large-scale Wireless Sensor Networks," ACM International Conference on Mobile Computing and Network (MOBICOM '02), pp 148 – 159, September 2002.
- [10] M. R. Garey, D. S. Johnson, "Computers and Intractability: A guide to the theory of NP-completeness," W. H. Freeman, San Francisco 1979.
- [11] G. Galbiati, F. Maffioli, A. Morzenti, "A short note on the approximability of the maximum leaves spanning tree problem," Information Processing Letters 52, pp 45 – 49, 1994.
- [12] S. Arora, C. Lund, R. Motwani, M. Sudan, M. Szegedy, "Proof verification and the hardness of approximation problems," Proceedings of the Thirty-third Annual IEEE Symposium on Foundations of Computer Science, pp 14 – 23, 1992.
- [13] S. Arora, S. Safra, "Probabilistic checking of proofs: A new characterization of NP," Proceedings of the Thirty-third Annual IEEE Symposium on Foundations of Computer Science, pp 2 – 13, 1992.
- [14] H. Lu, R. Ravi, "A near-linear time approximation algorithm for maximum leaf spanning tree," Journal of Algorithms, Vol. 29, No. 1, pp 132 – 141, 1998.
- [15] R. Solis-Oba, "2-approximation algorithm for finding a spanning tree with maximum number of leaves," Proceedings on the 6th Annual European Symposium on Algorithms, LNCS 1461, pp 441 – 452, 1998.
- [16] P. Bjorklund, P. Varbrand and D. Yuan, "Resource optimization of spatial TDMA in ad hoc radio networks," In Proceedings of the 2003 INFOCOM, San Francisco, CA, April1-3 2003.
- [17] J. Grönkvist, "Traffic controlled spatial reuse TDMA in multi-hop radio networks," Proceedings of 9th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, pp 1203 – 1207, 1998.

## Appendix: The Original 2-Approximation Algorithm for MLST

Let  $G = (V, E)$  be an undirected connected graph. The algorithm first builds a forest  $F$  by using a sequence of expansion rules, to be defined shortly. Then the trees in  $F$  are linked together to form a spanning tree  $T$ . Every tree  $T_i$  of  $F$  is built by first choosing a vertex of degree at least 3 as its root. Then the expansion rules described in Fig. 1 are used to grow the tree. If a leaf  $x$  has at least two neighbors not in  $T_i$  then the rule shown in Fig. 1(b) is used which places all neighbors of  $x$  not belonging to  $T_i$  as its children. On the other hand, if  $x$  has only one neighbor  $y$  that does not belong to  $T_i$  and at least two neighbors of  $y$  are not in  $T_i$ , then the rule shown in Fig. 1(a) is used. This rule puts  $y$  as the only child of  $x$  and all the neighbors of  $y$  not in  $T_i$  are made children of  $y$ . A tree  $T_i$  is grown until none of its leaves can be expanded.

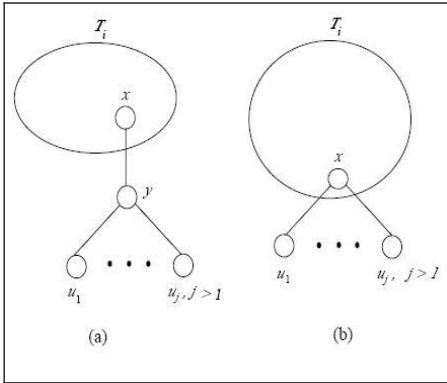


Fig. 1. Expansion rules

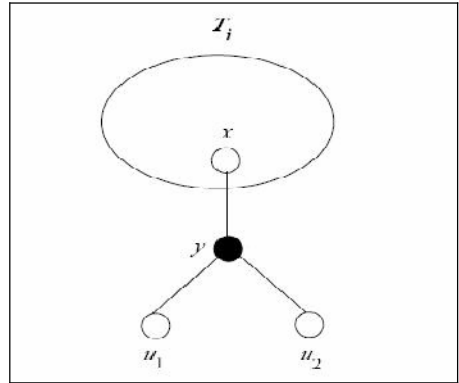


Fig. 2. The rule of priority 1

The expansion rules are assigned priorities as follows. The rule shown in Fig. 2, namely a leaf  $x$  has a single neighbor  $y$  not in  $F$  and  $y$  has exactly two neighbors outside  $F$ , has priority 1. All other expansion rules have priority 2. When building a tree  $T_i \in F$ , if two different leaves of  $T_i$  can be expanded, the leaf that can be expanded with the highest priority rule is expanded first. If two leaves can be expanded with rules of the same priority, then one is arbitrarily chosen for expansion. The full algorithm is shown in Fig. 3.

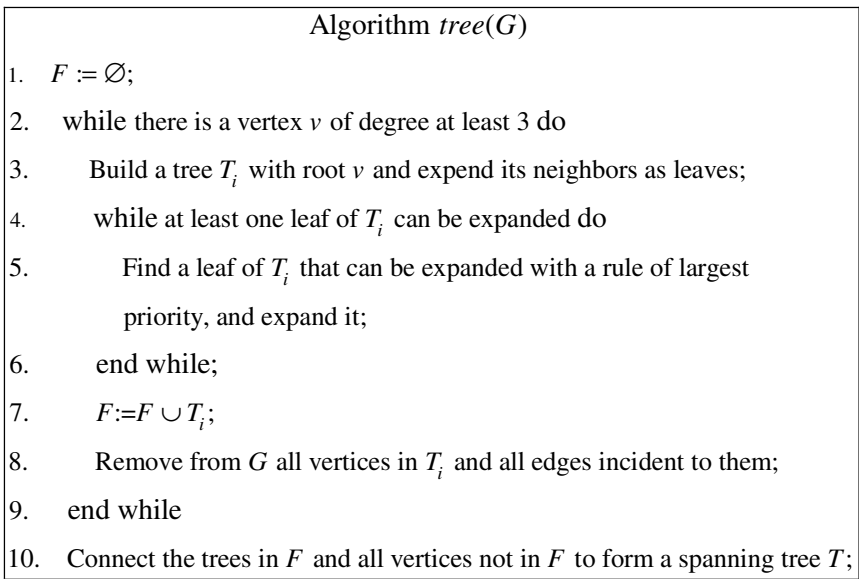


Fig. 3. The original 2-approximation algorithm for MLST



# Self Location Estimation Scheme Using ROA in Wireless Sensor Networks

Yun Kyung Lee, Eui Hyeok Kwon, and Jae Sung Lim

Graduate School of Information & Communications, Ajou University,  
Suwon, Geonggido 443-749, Republic of Korea  
{blue9337, k31001, jaslim}@ajou.ac.kr

**Abstract.** In wireless channel environments, location estimation error is inevitable when we use the ROA(Received signal strength Of Arrival) for location estimation. Because of multi-path fading and shadowing the received signal strength can not be obtained appropriately in wireless channel. Especially in NLOS(Non-Line Of Sight) environments, the location estimation error is increased significantly. Therefore, it is required to consider the wireless channel to reduce the estimation error. In this paper, we propose a new location estimation scheme to work precisely in NLOS environments. Reference nodes more than three are used to broadcast their location information continuously and each sensor node sorts the average received signal strength considering shadowing characteristics. We use the residual weighting algorithm[1] for ROA location estimation to obtain higher accuracy. As a result, we can obtain accurate location estimation only using location estimation scheme without any special device for location awareness.

## 1 Introduction

Recent growth of interest in pervasive computing and location aware system provides a strong motivation to develop the techniques for estimating the location of devices in both outdoor and indoor environments. There have been many approaches to solve this problem[2]. TOA(Time Of Arrival), TDOA(Time Of Difference Arrival), and ROA(Received signal strength Of Arrival) are location aware methods which calculate the relative distance between reference nodes and a sensor node.

TOA uses the time of received signals from the reference nodes to calculate distance. This method requires accurate time synchronization among all of sensor nodes and reference nodes. In case of TDOA, synchronized reference nodes receive signals from a sensor node and calculate time differences between times on which each reference node received signals from the sensor node. However, these methods are hard to be applied to wireless sensor networks since the time or time difference accuracy and synchronization are very sensitive to wireless channel especially in picocell environments.

Existing location aware method based on wireless sensor networks are Centroid[3], APIT[4], and DV-Hop/DV-Distance[5]. Centroid assumes that all signals from reference nodes have the same transmission range and the reference nodes are arranged regularly. So the regular arrangement of reference nodes makes radio transmission range overlapped. And each sensor node recognizes overlapped area as the location of

sensor node. Although this scheme makes location estimate easy it is difficult to estimate an exact location of a sensor node.

In APIT(Approximate Point In Triangle)[4], reference nodes send location information periodically and each sensor node makes triangles based on the received location information of reference nodes. In this process, a sensor node confirms whether its own location is inside of triangles using received signal strength. And the sensor node recognizes overlapped area of triangles as the location of the sensor node. The defect of APIT is that the precision of location estimate is low and needs a lot of reference nodes because APIT uses self developed method to estimate location. In DV-Hop/DV-Distance[5], reference nodes transmit their location information periodically and calculate average hop distances using received location information from other reference nodes. And they send the information of average hop distances to neighbor sensor nodes. Sensor nodes which receive this information calculate distance between reference nodes and aware their location. In DV-Distance, instead of transmitting average hop distances, they use signal strength between hops to calculate the distances between reference nodes. However, the location aware method used in DV-Hop/DV-Distance cannot guarantee accurate location aware.

In this paper we use an ROA method to obtain location awareness adopting the residual weighting algorithm[1] to reduce location estimation error. Besides reference nodes repeat broadcasting their location information and a sensor node categorizes average received signal strength(RSS) according to shadowing property to estimate accurate location awareness. In this method, we assume that signal power loss according to fast fading can be overcome through averaging of received signals. To estimate location a sensor node saves the received signal strengths from each reference nodes more than three. And the sensor node sorts the received signal strength from each reference node to the sensor node based on shadowing characteristics. And then the sensor node calculates the average distance between the reference nodes and the sensor node used by received signal strength which less effecting shadow. After doing range measurement the sensor node estimates its own location using the residual weighting algorithm[1].

The organization of this paper is as follows. In section 2, we explain the residual weighting algorithm using ROA for location estimation. Then simulation environments and simulation results are respectively investigated in section 3 and 4. Some conclusions are presented in section 5.

## 2 Residual Weighting Algorithm Using ROA

### 2.1 Range Measurement Scheme Using ROA

The observation space shown in Fig. 1 is a set of range measurements, and the parameters that need to estimate location are the geographical coordinates of the sensor node. We assume the random variation of RSS is a log normal Gaussian random variable due to shadowing effect. Thus we can describe that RSS in dB is distributed with  $X_{\sigma_s} = N(0, \sigma^2)$  of zero mean and  $\sigma^2$  variance like Fig. 2. In Fig. 2 each point means an averaged signal strength received from one of reference nodes to the sensor node. When a sensor node estimates an average signal strength from one of reference node

to the sensor node the sensor node uses the averaged signal strength which is less effecting shadow based on log normal Gaussian distribution and discards the remnant of the averaged received signal strength aren't used to estimate average signal strength. So we can define the ratio of averaged received signal strength samples the sensor node uses to estimate average signal strength. If the number of the averaged received signal strength samples the sensor node saves from one of reference nodes is 100 and the number of averaged received signal strength samples the sensor node uses to estimate average signal strength is 60 based on log normal Gaussian distribution the ratio of averaged received signal strength samples the sensor node uses to estimate average signal strength is 0.6.

In Fig. 1 the reference nodes(R1~R4) are repeatedly broadcasting data involving its own location information to the sensor node(X). After receiving the signals from each reference node the sensor node sorts the received signal strengths based on shadowing characteristics like Fig. 2 and estimates the average received signal strength used by received signal which is less effecting shadow. And then the sensor node calculates average distances( $d_1 \sim d_4$  in Fig.1) from each reference node to the sensor node through average received signal strength.

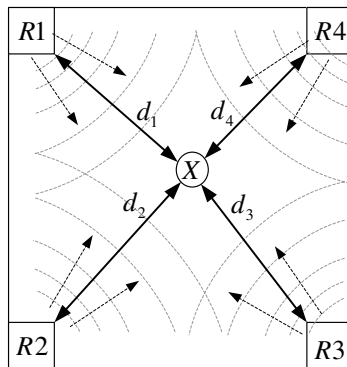


Fig. 1. Referenced Network Topology.  $R_i$  : Reference node  $i$ ,  $X$ : Sensor node.

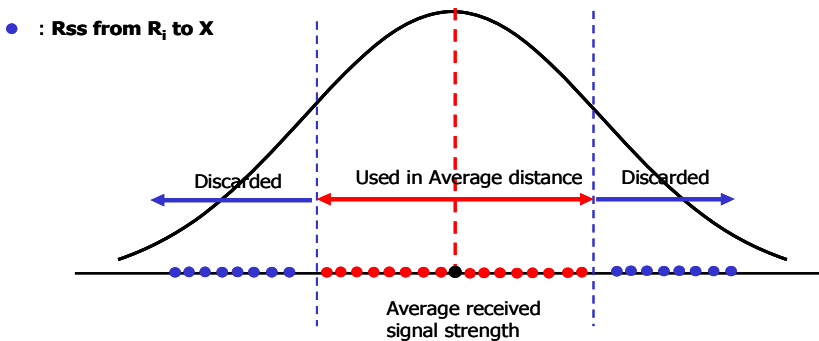


Fig. 2. Distribution of received signal strength

## 2.2 Residual Weighting Algorithm Using ROA

The residual weighting algorithm[1] was developed to protect location estimate from NLOS error corruption using TOA in the CDMA system. In this paper we use the algorithm to decrease ranging error when a sensor node does a self-location estimation using ROA in shadowing fading channels.

After range measurements using ROA the sensor node of X in Fig. 1 determines the location using LS(Least Square) estimator. That is

$$\hat{X} = \arg \min \sum_{i \in S}^N (|X - R_i| - d_i)^2 \tag{1}$$

where

- |X - R<sub>i</sub>| : distance between vectors x and R<sub>i</sub>
- X : [x,y]<sup>T</sup>, sensor node position in the Cartesian coordinate
- R<sub>i</sub> : [X<sub>Ri</sub>, Y<sub>Ri</sub>]<sup>T</sup>, reference node position in the Cartesian coordinate
- d<sub>i</sub> : the range measurement from the sensor node to the i-th reference node, i ∈ s
- N : the number of reference nodes
- S : the reference node index set

In (1) (|X - R<sub>i</sub>| - d<sub>i</sub>) is called the i-th residual for the sensor node X. And the range measurement of d<sub>i</sub> is not accurate due to the blocking of the direct path in shadow fading channel.

Equation (1) implies that the LS location estimator,  $\hat{X}$ , is an estimate which minimizes the sum of the residual squares over the data set, so (|X - R<sub>i</sub>| - d<sub>i</sub>) can define the residual of the i-th estimated location given by

$$R_{es}(\hat{X}; S) = \sum_{i \in S} (|\hat{X} - R_i| - d_i)^2 . \tag{2}$$

Thus, (1) could be rewritten as

$$\hat{X} = \arg \min R_{es}(X; S) \tag{3}$$

The LS location estimator,  $\hat{X}$ , is an estimate which minimizes  $R_{es}(X; S)$ . So that is

$$R_{es}(\hat{X}; S) = \min_X R_{es}(X; S) . \tag{4}$$

Equation (4) states that a good estimate is the one with minimum residual. When the sensor node does range measurement using ROA, ranging error is always occurred due to shadowing fading channel. So residual  $R_{es}(\hat{X}; S)$  would not be zero. In the paper we can use a residual  $R_{es}(\hat{X}; S)$  as the quality indicator of its estimate,  $\hat{X}$ .

To determine two dimensional location of the sensor node the minimum number of measurements based on ROA is required no less than three. The reference node index set(S) means the group doing range measurement in various ways subject to the constraint with equation (5). For example if the number of reference nodes is M = 4 (R1~R4) there are 5 eligible range measurement combinations.

1. Select 4 out of 4 :  $\binom{4}{4} = 1$  combination
2. Select 3 out of 4 :  $\binom{4}{3} = 4$  combinations

$$N = \sum_{i=3}^M \binom{M}{i} \tag{5}$$

Therefore there are 5 different range measurement combinations. Applying LS estimator on these combinations using equation (1), we can obtain 5 sensor node estimates,  $\hat{X}$ , which are denoted as intermediate location estimates.

To use  $R_{es}(\hat{X}; S)$  as the quality indicator of its estimates,  $\hat{X}$ , it needs to match the numbers of range measurements in the groups. So the normalized  $R_{es}(\hat{X}; S)$  is defined as  $\tilde{R}_{es}(\hat{X}; S)$  given in (6) to remove the dependence on the size of the group.

$$\tilde{R}_{es}(\hat{X}; S) = \frac{R_{es}(\hat{X}; S)}{m} \tag{6}$$

where  $m$  is the size of  $S$ . If  $\hat{X}$  is obtained by using four reference node in equation (1) the value of  $m$  is four and by using three reference node in equation (1) the value of  $m$  is three.

Finally, the location of sensor node of  $X$  is obtained through equation (7) as the weighted linear combination of the intermediate estimates,  $\hat{X}_{1..N}$  with its  $\tilde{R}_{es}(\hat{X}_{1..N}; S)$ . The weight is inversely proportional to  $\tilde{R}_{es}(\hat{X}; S)$  of the estimate. It means that we can decrease ranging error generated by shadow fading as we could rely more on the estimates derived from those groups which estimates used by signal strengths impacted less effecting shadow.

$$\hat{X} = \frac{\sum_{k=1}^N \hat{X}_k (\tilde{R}_{es}(\hat{X}_k; S_k))^{-1}}{\sum_{k=1}^N (\tilde{R}_{es}(\hat{X}_k; S_k))^{-1}} \tag{7}$$

### 3 Simulation Environments

To generate RSS samples as a function of distance the path loss model with the log-normal shadowing effects is used[7]:

$$PL(d) = 30 + 21 \log_{10} \left( \frac{d}{d_0} \right) + X_{\sigma_s} \tag{8}$$

where  $PL(d)$  is the path loss for the distance between reference nodes and the sensor node. To consider indoor environment in the simulation we assumed that the value of  $d_0$  is 1m, the path loss exponent ( $\eta$ ) is 2.1 and path loss for a reference distance is 30. And the transmit power ( $P_T$ ) of reference nodes is fixed as 10dBm. The random variation of RSS in dB is expressed as a Gaussian random variable of zero mean and variance of  $\sigma^2$ . In order to generate shadowing effects the value of standard deviation is used 7dB in LOS condition and 9.7dB in NLOS condition.

To generate NLOS measurements NLOS errors are inserted to range measurements in addition to the measurement errors. In this simulation the exponential distribution with mean 1.5dB ~2dB as NLOS error model is used. All powers are expressed in dBm and all distances in meters.

The system model for simulation is Fig. 1. We evaluated the performance of the residual weighting algorithm using ROA in wireless sensor networks when received signals impacted less shadow effects are used. Four reference nodes are broadcasting their location information repeatedly. And a sensor node classifies received signal strengths based on shadowing characteristics and calculates the distances from each reference node to the sensor node using average received signal strength which is obtained by received signal strengths impacted less shadow effects. And the number of samples which a reference node is broadcasting are 20 ~ 100.

The performance of residual weighting algorithm(Rwgh) using ROA was compared with that of LS estimator(LSE) which is obtained the location of sensor node by only equation (1) using the distances through ROA from each reference node to sensor node. Table 1 is the simulation parameters and ranges for the performance evaluation and ranges.

We used the MATLAB as simulator and the performance criteria of the algorithms as the root mean square error(RMSE) given by

$$RMSE = \sqrt{\frac{e^2_1 + e^2_2 \dots + e^2_n}{n - 1}} \tag{9}$$

where  $n$  is the number of trials estimating the location of the sensor node and  $e_{1...n}$  is the location error as Euclidean distances between the location estimate and the actual location of the unknown sensor node.

**Table 1.** Typical values and ranges of simulation parameters

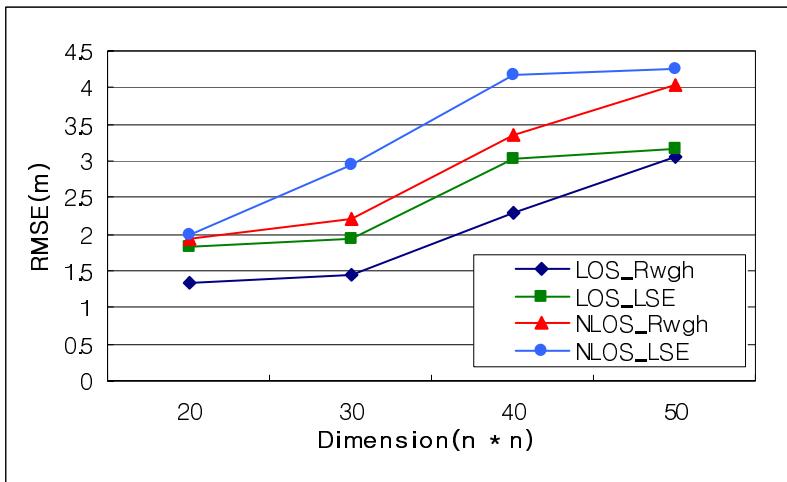
Parameters	Typical Value	Typical Range
$P_T$	10dBm	NA
$P_L(d_0)$	30dB	NA
$\eta$	2.1	NA
$\sigma_s$ (LOS)	7dB (indoor)	2-14 [7]
$\sigma_s$ (NLOS)	9.7dB(indoor)	2-14 [7]
Dimension	25m *25m	{50,40,30,20}
Node placement	Random	NA

## 4 Simulation Results

Fig. 3 shows the RMSE according to different dimensions where transmit power is 10dBm, the number of samples which reference nodes is broadcasting is 100, the values of standard deviation are 7dB in LOS condition and 9.7dB in NLOS condition for shadowing effects, and the value of mean of NLOS error model is 2dB. The ratio of averaged received signal strength samples the sensor node uses to estimate average signal strength is 0.6.

The results show that the value of RMSE is strongly affected by the increase of dimensions due to using ROA to calculate range measurement. In 25m\*25m the value of RMSE can be maintained below 2m in Rwgh algorithm regardless of LOS and NLOS conditions. It should note that the accurate location estimation can be obtained by ROA in wireless sensor networks.

And in 40m\*40m it shows that LOS\_Rwgh algorithm is more stable than LOS\_LSE because the value of RMSE doesn't exceed 2.5m in LOS\_Rwgh algorithm although received signal strength is week against dimension.



**Fig. 3.** RMSE according to increase of dimensions

Fig. 4 shows the RMSE for shadowing effects with the different values of standard deviations {2, 4, 6, 8, 10, 12, 14} where transmit power of reference node is 10dBm, the dimension is 25m\*25m, the number of samples which reference nodes is broadcasting is 100, the ratio of averaged received signal strength samples the sensor node uses to estimate average signal strength is 0.6 and the value of mean of NLOS error model is 1.5dB.

The results show that the value of RMSE doesn't exceed about 2m with changes of the value of standard deviation in Rwgh. So it means that we can get accurate location estimation in LOS condition and NLOS condition using ROA.

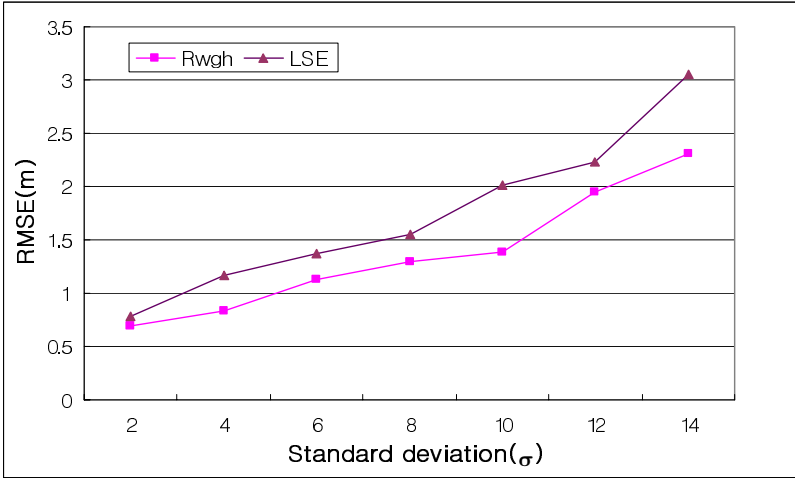


Fig. 4. RMSE according to changes of standard deviations

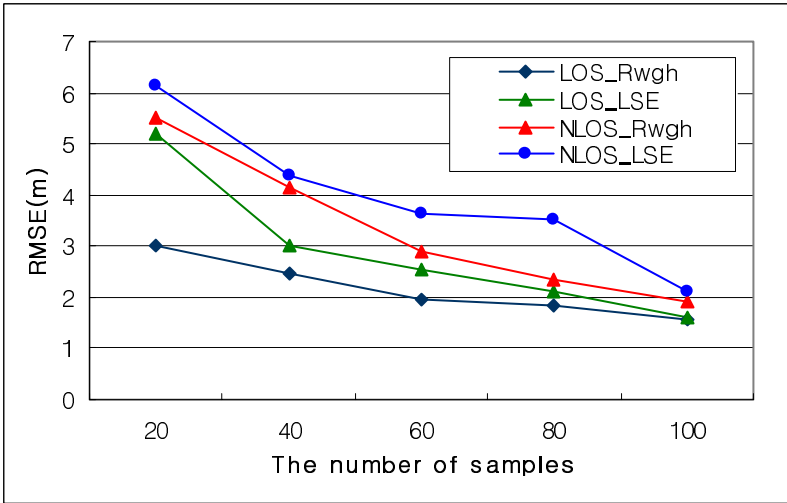


Fig. 5. RMSE for different number of samples

Fig. 5 shows the RMSE for different number of samples. Each RSS sample is generated every packet. In the simulation the dimension is 25\*25 and the ratio of averaged received signal strength samples the sensor node uses to estimate average signal strength is 0.8.

The results show that LOS\_Rwgh is more stable than LOS\_LSE because the slope of graph in LOS\_Rwgh changes slowly. When the number of samples which reference nodes is broadcasting are less than 60 the values of RMSE in NLOS\_Rwgh increase almost 4m.



## 5 Conclusions

In this paper we proposed a self location estimation scheme using ROA for wireless sensor networks. We can observe through the simulation results that the accurate location estimation can be obtained in LOS/NLOS condition in wireless sensor networks (not exceeding 2.5m at dimension of 25m\*25m). And it is possible to get accurate location estimation only using location estimation scheme without any special device for location awareness. But lots of RSS samples are required and the reference nodes should broadcast repeatedly to get more accurate location estimation.

As the future work we are developing a mean value estimation of RSS to obtain high accurate location estimation and an MAC protocol suitable for transmitting location information efficiently.

## Acknowledgements

This work was supported in part by the Ministry of Commerce, Industry and Energy, Korea, grant No.10023210 and in part by grant No. R01200300010724 from the Basic Research Program of the Korea Science & Engineering Foundation.

## References

1. Pi-Chun Chen: A Non-line of Sight Error Mitigation Algorithm in Location Estimation, IEEE Wireless Communications and Networking Conference, vol. 1, pp. 316–320, 1999.
2. Jochen schiller and Agnes voisard: Location- Based Services, Morgan Kaufmann.
3. N. Bulusu and J. Heidemann: GPS-less Low Cost Outdoor Localization for very small devices, IEEE Personal Communications Magazine, October 2000.
4. Tian He, Chengdu Huang, B. M. Blum, and John A. Stankovic: Range Free Localization Schemes in Large Scale Sensor Networks, Mobicom 2003.
5. D. Niculescu and B. Nath: DV Based Positioning in Ad hoc Networks, Telecommunication Systems, Kluwer, 22(1-4), pp. 267-280, January-April 2003.
6. Asis Nasipuri and Kai Li: A Directionality based Location Discovery Scheme for Wireless Sensor Networks, Wireless Sensor Networks and Applications, September 2002.
7. Rappaport: Wireless Communications principles and practice, Prentice Hall.
8. H. Hashmi: The indoor radio propagation channel. Proceeding of the IEEE, 81(7), pp. 943-968, July 1993.
9. F. Mondinelli and Zsolt: Self-Localizing Sensor Network Architectures, IEEE Transactions on Instrumentation and Measurement, vol. 53, no. 2, April 2004.
10. David Christopher Moore: Robust Distributed Sensor Network Localization with Noisy Range Measurements, California Institute of Technology, The ACM conference on Embedded Networked Sensor Systems, November 2004.

# Energy-Efficient Target Localization Based on a Prediction Model\*

Yu Gu, Wei Zhang, HengChang Liu, Baohua Zhao, and Yugui Qu

Department of Computer Science, University of Science and Technology of China,  
Hefei, Anhui 230027, China

Laboratory of Computer Science Institute of Software Chinese Academy of Sciences  
{guyu, , hcliu}@mail.ustc.edu.cn  
{bhzhao, ygqu}@ustc.edu.cn

**Abstract.** In this paper, we propose an energy-efficient target localization scheme (TLS) based on a prediction model that can reveal the most likely zone the target will be in, and also a corresponding two-step communication protocol between base station (BS) and sensors. BS uses a query mechanism to determine which sensors should be used for detailed information according to a limited amount of data received from the sensors. This scheme reduces both energy consumption and communication bandwidth requirement, prolongs the lifetime of the wireless sensor networks. Simulation results indicate that it can achieve a high accuracy while saving a large amount of energy.

**Keywords:** wireless sensor networks, energy-efficient, prediction model, TLS.

## 1 Introduction

Many sensor networks need to handle physical entities that move in the environment (e.g. habitat monitoring [1] and intruder tracking [2]). Only sensors close to the target should participate in the aggregation of data associated with that target, since activating distant sensor nodes wastes precious energy without improving sensing fidelity. In recent research, a reliable multicast method has been proposed to solve this kind of problem in literature [3] and achieved comparatively good results. Since energy management in the localization event is crucial for battery-driven sensor nodes where are severely energy constrained, here we focus on reducing energy consumption in wireless sensor networks for target localization.

In general, a sensor network has an almost constant rate of energy consumption if no target activities are detected in the sensor field. The minimization of energy consumption for an active sensor network with target activities is more complicated since target detection involves collaborative sensing and communication with different sensors.

---

\* This paper is supported by the National Natural Science Foundation of China under Grant No. 60241004, the National Grand Fundamental Research 973 Program of China under Grant No. 2003CB314801, and the State Key Laboratory of Networking and Switching Technology.

Though many literatures [4] [7] [8] [9] [11] have already introduced different methods to save energy during tracking procedure, without prediction knowledge of the target, most of energy consumed in the localization event still has been wasted in keeping too many sensors in the sensing state while actually only a few sensors will be enough.

To continuously monitor a mobile entity, a sensor network must maintain an active group that moves at the same velocity as the entity. This energy-efficient operation requires a prediction model to make conjectures about future target positions. Here in this paper, an energy-efficient target localization scheme (TLS) based on a prediction model has been posted. Since estimating the accurate future positions is almost impossible, the prediction model uses previous target position information to reveal the most likely zone (We call this zone as an awaken zone (AZ)) target will be in at the next moment, and then BS sends commands to wake up enough sensors inside AZ to track the target. We will show how to calculate AZ at time  $i$  and prove that there will always be some sensors in AZ that can detect the target under some assumptions. To the maximum of my knowledge, no previous work has been done on the exact prediction model of saving energy.

Two distinct improving results have been obtained:

- This scheme reduces averagely 80% energy consumption during the track procedure as the simulation shows.
- Analysis on TLS indicates that the requirement of communication bandwidth can also be reduced since the packets generated in the communication protocol are all kept very small.

In the past few years, a number of detecting and tracking methods have been proposed [4] [5] [6] [7] [8] [9] [10] [11] [12]. Those literatures focus either on how to reduce energy consumption or on how to localize a target more accurately.

Yi Zou and Krishnendu Chakrabarty have proposed an energy-aware target detection and localization strategy for cluster-based wireless networks in [4]. This strategy is based on a posteriori algorithm with a two-step communication protocol between cluster head and the sensor within the cluster. It uses a detection probability table which contains entries for all possible detection reports from those sensors that can detect a target at grid points to find out the target's max-probability position. In [7], T.Clouqueur and his copartners have discussed sensor deployment for collaborative target detection where path exposure is used as a measure of the effectiveness of the sensor deployment. Richard R. Brooks and his copartner have proposed a formulation which is anchored on location-aware data routing to conserve system resources, such as energy and bandwidth [9]. Literature [12] models the problem of a sensor network tracking a moving target as a Markov model of mobility, that is, the position of the target at time  $i$  depends only on its position at time  $i-1$ .

Range-based protocols use absolute point-to-point distance or angle information to calculate location between neighboring sensors. Common techniques for distance/angle estimation include Time of Arrival (TOA) [14], Time Difference of Arrival (TDOA) [13], Angle of Arrival (AOA) [15], and Received Signal Strength (RSS) [13]. Those protocols all can work well in our scheme.

Considering that sensor density inside some AZ may be too high, we only need to wake up enough sensors in AZ instead of all sensors. Here we use the ASCENT method proposed in [16].

The remainder of the paper is organized as follows. Section 2 introduces system model and a few consumptions. Section 3 describes details about TLS. Section 4 shows energy consumption analysis. In section 5, we present simulation results and our practical considerations. Section 6 concludes the paper and outlines directions for future work.

## 2 System Model and Assumptions

In this section, we will introduce several assumptions in part A. In part B, a prediction model is described, and an exact definition of AZ is given with the existence of sensors inside that will detect the target in the future moment proven.

### A. Assumptions

For a simple analysis, we make the following assumptions in this paper:

- All sensors remain stationary after deployment phase.
- Sensors are able to communicate with BS, while BS has the exact deployment information.
- BS is much more powerful in computation capability than sensors. BS is responsible for all calculation, while sensors are functioning mainly as data collecting devices.
- Considering a wireless sensor network with N sensors deployed averagely in dual space, we regard it reasonable to assume that there are k sensors per  $\pi r_s^2$  ( $k \geq 1, k \in Z^+$ ), ( $r_s$  represents the detection range of sensors).

### B. Prediction Model

In this part, we introduce the prediction model for target localization. A key contribution of this model is that it supplies BS with the information of an area where the target will be at the next moment. If the speed of a target moving in a wireless sensor network is lower than a threshold (We will show that the threshold in actual applications can be set high enough in Section 6), we can prove that there is at least one sensor inside that area which can detect the target.

To predict the target's position at time i, we should use the previous position information for target trace is always continuous in an actual environment.

**Definition 1.**  $\bar{I}$ ,  $\bar{I}$  is the target trace.  $\bar{I} = (I_0(x_0, y_0), \dots, I_i(x_i, y_i), \dots)$ ,  $I_i(x_i, y_i)$   $i \geq 0$  represents the target's position at time i.

Given the target trace from time 0 to time i-1:  $\bar{I} = (I_0(x_0, y_0), \dots, I_{i-1}(x_{i-1}, y_{i-1}))$ , we model target's position at time  $i(i \geq 1)$  following a two-dimensional Gaussian distribution, which is centered at the prediction point  $(x_{pre-i}, y_{pre-i})$ . Namely, the

mean of the Gaussian distribution  $\mu$  equals  $(x_{pre-i}, y_{pre-i})$ , and the probability distribution function (PDF) for the target's position at time  $i$  is the following:

$$f(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x-x_{pre-i})^2+(y-y_{pre-i})^2}{2\sigma^2}} \tag{1}$$

(Here  $\sigma$  is the standard deviation). Without loss of generality, we assume that the standard deviation for all PDF is identical.

The prediction point  $(x_{pre-i}, y_{pre-i})$  represents the most possible position of the target at time  $i$ . It depends on the prior information of the target trace. Here we apply two different methods according to different situations in the practical applications.

I. Suppose the target walks along the line:  $y = y_{i-1} + (x - x_{i-1}) \tan \theta$ , then we can define the prediction point as: ( $\Delta t$  is the time interval between time  $i-1$  and  $i$ )

$$\begin{cases} x_{pre-i} = x_{i-1} + V_{i-1}\Delta t \cos \theta \\ y_{pre-i} = y_{i-1} + V_{i-1}\Delta t \sin \theta \end{cases} \quad (i \geq 2 \quad \theta = \arctan(\frac{y_{i-1} - y_{i-2}}{x_{i-1} - x_{i-2}})) \tag{2}$$

$V_{i-1}$  represents target's instantaneous speed at time  $i-1$ :

$$V_{i-1} = \frac{|I_{i-1} - I_{i-2}|}{\Delta t} \quad i \geq 2 \tag{3}$$

II. If we do not have much useful prior information of the target, which is common in actual environment, then the target position at time  $i-1$  would be used as the prediction point:

$$\begin{cases} x_{pre-i} = x_{i-1} \\ y_{pre-i} = y_{i-1} \end{cases} \quad i \geq 1 \tag{4}$$

Since it is difficult to estimate future target position accurately, here instead of predicting the target position at time  $i$ , we will predict an area large enough in which the target will be by using (1).

According to (1), if the target trace  $\vec{I} = (I_0(x_0, y_0), \dots, I_{i-1}(x_{i-1}, y_{i-1}))$  is known, the probability that target will be in the round centered at prediction point  $(x_{pre-i}, y_{pre-i})$  with radius  $R$  that we can calculate using Theorem1 (introduced below) is greater than a given threshold  $H(0 \leq H \leq 1)$ . It means that we can calculate an area in which the target will be at time  $i$  with a probability greater than  $H$ .

**Theorem 1.** Given a threshold  $H(0 \leq H \leq 1)$ ,  $\exists R \geq \sigma \sqrt{2 \ln \frac{1}{1-H}}$ , the probability that target will be in the round centered at prediction point  $(x_{pre-i}, y_{pre-i})$  with radius  $R$  at time  $i$  is larger than  $H$ .

**Proof:** See Appendix A. □

We have mentioned before that BS would wake up enough sensors in an awaken zone (AZ) for future position detection, and an exact definition of AZ is given below:

**Definition 2.** (Awaken Zone-i) Awaken Zone-i is a round centered at point  $(x_{pre-i}, y_{pre-i})$  with radius  $R = \sigma \sqrt{2 \ln \frac{1}{1-H}}$ . BS sends command to wake up enough sensors inside AS-i to keep sensing target. (We will use AZ instead of AZ-i in the following context)

Although the threshold  $H$  cannot be preset in the paper, we still suggest that in practical applications  $H$  should be larger than 90% to make sure enough sensors in AZ can detect the target.

In the following part, we will approve the existence of at least one sensor in AZ we defined could detect the target at time  $i$ .

**Definition 3.**  $V_H \cdot V_H$  is the maximum speed a target can have if it can be detected by sensors in AZ.  $V_H = \frac{l_{max}}{\Delta t}$ , where  $l_{max}$  is the result of equations following: ( $r_s$  represents the detection range of sensors):

$$\begin{cases} S = r_s^2 \theta_1 - r_s^2 \sin \theta_1 \cos \theta_1 + R^2 \theta_2 - R^2 \sin \theta_2 \cos \theta_2 \\ R = \sigma \sqrt{2 \ln \frac{1}{1-H}} \\ S * k / \pi r_s^2 = 1 \end{cases}$$

$$\theta_1 = \cos^{-1} \left( \frac{r_s^2 + l_{max}^2 - R^2}{2 r_s l_{max}} \right) \quad \theta_2 = \cos^{-1} \left( \frac{R^2 + l_{max}^2 - r_s^2}{2 R l_{max}} \right)$$

**Theorem 2.** According to our pervious assumption: there are  $k$  sensors per  $\pi r_s^2$  averagely. ( $k \geq 1, k \in Z^+$ ), if  $V_i \leq V_H$ , there will be at least one sensor inside Awakening Zone-i that can detect the target at time  $i$ .

**Proof:** See Appendix A □

In this section, we have introduced our sensor nodes mode, prediction model and an exact definition of AZ has been given including how to calculate an AZ-i. We have also proven that the existence of at least one sensor inside the AZ-i can detect the target at time  $i$  under some assumptions which will be discussed in Section 6.

### 3 Target Localization Scheme (TLS)

First we will discuss the common flow about TLS in this section, and then we will show detailed information about data types used in the communication protocol in part A. Since we apply distributed algorithms in TLS, two different types of timer shown in

part B are needed in the protocol. To save precious energy, after gathering draw date from sensors that have detected the target at current time, TLS uses a query mechanism to determine which sensors should be queried for further information which will be described in part C.

### A. Data Types

There are four different data types in our communication protocol.

**AWK:** when a sensor node in idle state receives an AWK packet, it wakes up, sets up a KEEPALIVE timer and keeps sensing for a constant time  $\Delta t$ . If it is already in sensing state, then the KEEPALIVE timer would be reset. An AWK packet is very small; actually, one bit will be enough.

**RPT:** when a sensor node detects an object, it generates a RPT packet indicating rest energy it has and the distance between them then sends it to BS.

**QUY:** BS sends a QUY packet to a sensor node, and waits for a DAT packet. A query message can also be encoded into a few bits.

**DAT:** When a sensor node receives a QUY packet, it sends a DAT packet to BS. A DAT packet includes detailed information about the target.

Since all the packets generated in our communication protocol are very small, in fact all the information can be encoded into only a few bits, TLS could reduce bandwidth requirement and the latency as well.

### B. KEEPALIVE Timer and WAIT Timer

When a sensor in the idle state receives an AWK packet, it wakes up and set a KEEPALIVE timer. While the KEEPALIVE timer does not expire, it keeps sensing. But if a sensor already in the sensing state receives an AWK packet, it would reset the timer and keep sensing.

After BS receives a RPT message for time  $i$ , it sets a WAIT timer which indicates how long BS should wait for all RPT messages. When WAIT timer expires, BS executes Query Mechanism to query sensors for detailed information. We can set  $\frac{3\Delta t}{4}$  as its value.

### C. BS Query Mechanism

To save energy, we only need to query  $k_q(i)$  sensors for detailed information at time  $i$ . When BS receives a REQ packet, if it is the first time to receive a detection report for time  $i$ , BS sets a WAIT timer. After the WAIT timer expires, BS considers that all sensor nodes that detect the target at time slot  $\Delta t$  between time  $i-1$  and time  $i$  have been reported. Then it uses Query Mechanism to select sensors to query.

Suppose there are  $k(i)$  ( $k(i) \geq k_q(i)$ ) sensors which have sent  $RPT < d, E_r >$  packets to BS at time  $i$ , ( $d$  represents the distance between the sensor and the target,  $E_r$  represents rest energy of the sensor,  $r_s$  represents the sensing range). We build the following equation:

$$F = \alpha \frac{E_r}{E} + (1 - \alpha) \frac{d}{r_s} \quad 0 \leq \alpha \leq 1 \tag{5}$$

BS must query sensor nodes that have larger F for further detailed information. Obviously, to save precious energy, we must query the sensor nodes which have the most rest energy. But if we want to achieve a higher accuracy, we should query the sensor nodes which have a nearer distance to the target. So we build this equation considering both the energy and accuracy to decide which sensors to query.

In this section, we have described detailed information about our scheme, and in Section 5, we will discuss the advantages of TLS in energy consumption.

### 4 Energy Consumption Analysis

In this section, to show TLS's impact on energy consumption, we will compare energy consumption between two different tracking schemes: Tracking an object in WSN without TLS and with TLS. A simplified sensor energy consumption model is used here as a metric for evaluating energy consumption. Suppose a sensor has three basic energy consumption types: sensing, active, idle, and these power values are  $\varphi_s, \varphi_a, \varphi_i$ .

**Definition 4.**  $T_r$ , if some sensor node detects the target, it needs to send a RPT packet  $\langle d, E_r \rangle$  to BS,  $T_r$  represents the time needed to transfer this peculiar packet. Since usually the length of this packet is constant, we assume that  $T_r$  is a constant value.

**Definition 5.**  $T_s$ , if some sensor node has received a QUY packet from BS, it needs to send BS a DAT packet. We define  $T_s$  as the time the sensor used to receive this QUY packet plus the time it needs to transfer a DAT packet to BS. It is also reasonable to assume  $T_s$  as a constant value.

#### A. Energy Consumption Analysis for Tracking an Object in WSN Without TLS

Without available prediction information, we must keep all sensor nodes in sensing state to make sure the target can be detected. Assume there are  $k(i)$  sensors that can detect the target at time i, but only  $k_q(i) (k_q(i) \leq k(i))$  sensors would be queried by BS for detailed information.

In time slot  $\Delta t$  between time i-1 and i:

Number of sensors in sensing state: N

Number of sensors that need to generate RPT packets:  $k(i)$

Number of sensors that need to generate DAT packets:  $k_q(i)$

We evaluate the energy consumption as following:



$$E_i = N\varphi_s\Delta t + k_q(i)\varphi_aT_s + k(i)\varphi_aT_r \tag{6}$$

$$E = \sum_{i=0}^{i_{end}} E_i \tag{7}$$

**B. Energy Consumption Analysis for Tracking an Object in WSN with TLS**

In time slot  $\Delta t$  between time  $i-1$  and  $i$ , considering that there are  $k^{\wedge}(i)$  ( $k^{\wedge}(i) \leq k(i)$ ) sensors that can detect the target, but only  $k_q(i)$  ( $k_q(i) \leq k(i)$ ) sensors would be queried by BS for detailed information.

Number of sensors in the sensing state:  $k(\frac{\sigma}{r_s})^2 \ln \frac{1}{1-H}$

Number of sensors that detect the target:  $k^{\wedge}(i)$  ( $k^{\wedge}(i) \leq k(i)$ )

Number of sensors that need to generate DAT packets:  $k_q(i)$

Number of sensors in the idle state:  $N - k(\frac{\sigma}{r_s})^2 \ln \frac{1}{1-H}$

Energy consumption using TLS can be expressed as

$$E_i^{\wedge} = k(\frac{\sigma}{r_s})^2 \ln \frac{1}{1-H} \varphi_s\Delta t + k_q(i)\varphi_aT_s + k^{\wedge}(i)\varphi_aT_r + (N - k(\frac{\sigma}{r_s})^2 \ln \frac{1}{1-H})\varphi_i\Delta t$$

$$E = \sum_{i=0}^{i_{end}} E_i^{\wedge} \tag{8}$$

Therefore, the difference consumption  $\Delta E_i = E_i - E_i^{\wedge}$  can be expressed as:

$$\Delta E_i = (N - k(\frac{\sigma}{r_s})^2 \ln \frac{1}{1-H})\varphi_s\Delta t + (k(i) - k^{\wedge}(i))\varphi_aT_r - (N - k(\frac{\sigma}{r_s})^2 \ln \frac{1}{1-H})\varphi_i\Delta t$$

$$= (N - k(\frac{\sigma}{r_s})^2 \ln \frac{1}{1-H})(\varphi_s - \varphi_i)\Delta t + (k(i) - k^{\wedge}(i))\varphi_aT_r$$

$$\Delta E = \sum_{i=0}^{i_{end}} \Delta E_i \tag{9}$$

Since  $\varphi_s < \varphi_i$  &  $k(i) \geq k^{\wedge}(i)$  (Because AZ is a part of the sensor field), we can see that the energy consumption is greatly reduced with the passage of time.

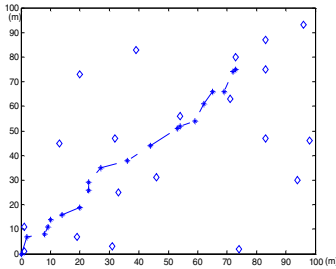
The comparison between tracking flows with TLS and without TLS has been shown in this section. We can see that the energy consumption is greatly reduced through equation (9) if using TLS. In the next section, we will show simulation results and practical consideration.

**5 Simulation and Practical Consideration**

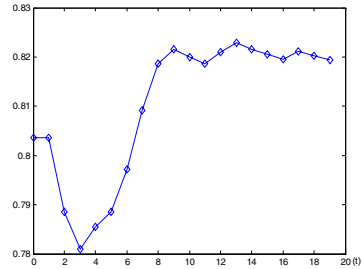
We present results for a case study carried out using MatLab6.1 and C++ in this section. The simulation is done on a 100m by 100m field grid with 20 sensors randomly placed in the sensor field.

The parameters of the prediction models are  $\sigma = 10, \Delta t = 0.3s, H = 90%$  ,  $r_s = 15m$  . We choose the energy consumption model parameters as  $\varphi_a = 400nj/s, \varphi_i = 100nj/s$  and  $\varphi_s = 1000nj/s$  , we have no physical data available for  $T_r$  and  $T_s$  ; however, their values do not affect the target localization procedure, therefore we only need to set them manually to satisfy the relationship  $T_r < T_s$  . In this case,  $T_r = 10ms, T_s = 100ms$ .

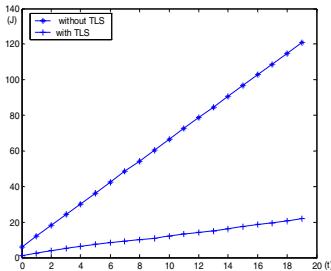
The layout of the sensor field is given in Fig 2, with a target trace randomly generated in the sensor field. The target travels from the position (0, 0). We assume the target locations are updated at discrete time instants in unit of seconds, and the granularity of the time is long enough for sampling by two neighboring locations in the target trace with negligible errors. For simplicity, we have evaluated TLS for  $k_q(i) = 1$  . Fig 3 presents the instantaneous energy saving in percentage, and Fig 4 presents the energy consumption for the case study as the target moves along its trace in the sensor field. From Fig 3 and Fig 4, we note that a large amount of energy is saved during target localization. Averagely 80% energy could be saved if TLS has been used.



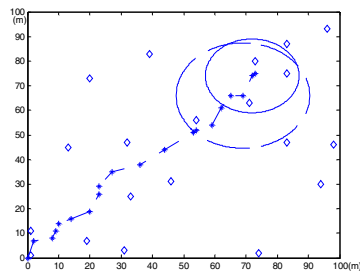
**Fig. 1.** Sensor Field with Target Trace



**Fig. 2.** Instantaneous Energy Saving in percentage



**Fig. 3.** Energy Consumption Comparison



**Fig. 4.** Prediction Model Analysis

Fig 5 shows how our prediction model works: there are four sensors needed to be woken up inside the dashed round that presents the AZ-19, while actually three of them

which are in the smaller round can detect the target at time 19. In our simulations, the prediction model has worked efficiently. (Here we use (4) to generate prediction points).

Practical consideration: In actual environment, according to our definition 3, we know that  $V_H$  follows:

$$\frac{R}{\Delta t} < V_H < \frac{R + r_s}{\Delta t} \quad (10)$$

Let us use some reasonable, common data to check  $V_H$ 's lower limit and upper limit, e.g:  $H=0.9$

$$\begin{aligned} \sigma = 10 &\Rightarrow R = 21m, \Delta t = 0.3s, r_s = 40m \\ &\Rightarrow 70m/s < V_H < 200m/s \\ &\Rightarrow 252km/h < V_H < 720km/h \end{aligned}$$

We can see that  $V_H$  is large enough for the target we track in the actual applications.

The standard deviation  $\sigma$  in (1) can be gained using transcendental information about the target in actual applications, and we believe that  $\sigma$  is proportion to  $V_H$ .

The simulation results shown in this section indicate that TLS runs smoothly in the tracking procedure while reducing energy consumption by 80%.

## 6 Conclusion and Future Work

In this paper we bring forward an energy-efficient target localization scheme named TLS based on a prediction model for wireless sensor networks. The TLS make use of previous target position information and PDF knowledge of the target's trace to reveal which area the target will be in at the next moment, then a limited amount of sleeping sensors inside that area will be woke up to keep alert. We also propose a corresponding two-step communication protocol between sensors and BS. The analysis shows that TLS can effectively reduce energy consumption as the simulation results indicate. Since packets transferred in the procedure are kept very small, TLS decreases the latency for target localization as well.

In our future work, we will focus on improving the accuracy of the TLS. If the track knowledge mode we use cannot accurately model the actual track, there will be extra errors in the localization. We will study the properties of this kind of errors in our future work and try to propose an error-smooth method.

## References

- [1] A.Cerpa, J. Elson, D.Estrin, L.Girod, M.Hamilton, and J.Zhao "Habitat monitoring: Application driver for wireless communications technology" in ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean, Costa Rica, April 2001.
- [2] D.Li, K Wong, YH.Hu, and A.Sayeed "Detection, classification and tracking of targets in distributed sensor network" IEEE Signal Processing Magazine, Vol. 19, no. 2, March 2002.

- [3] Qingfeng Huang, Chenyang Lu and Gruia-Catalin Roman “Reliable Mobicast via Face-Aware Routing” In Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2004), March 2004.
- [4] Yi Zou and Krishnendu Chakrabarty “Energy-Aware Target Localization in Wireless Sensor Networks” Proc IEEE International Conference on Pervasive Computing and Communications (PerCom 2003) pp. 60-67, 2003.
- [5] Dragoş Niculescu "Positioning in Ad Hoc Sensor Networks" IEEE Network Magazine, July/August 2004. pp. 24-29.
- [6] Hussam Al-Hertani and Jacek Ilow “Detection and Localization in a Wireless network of Randomly Distributed Sensors” CCGEI 2003, Montreal, May 2003.
- [7] T.Clouqueur, V.Phipatanasuphorn, P.Ramanathan, and K.K.Saluja “Sensor Deployment Strategy for Target Detection”, Proc. 1st ASM International Workshop on Wireless Sensor Networks and Applications, pp.42-48, September 2002.
- [8] J.Liu, P.Cheung, L.Guibas, and F.Zhao “A Dual-Space Approach to Tracking and Sensor Management in Wireless Sensor Networks” Proc. 1st ACM International Workshop on Wireless Sensor Networks and Applications, pp. 131-139, September 2002.
- [9] Richard R.Brooks, Parameswaran Ramanathan and Akbar M. Syeed “Distributed Target Classification and Tracking in Sensor Networks” Proceeding of the IEEE, Vol.91, No.8.August 2003
- [10] Ruixin Niu and Pramod K.Varshney “Target Location Estimation in Wireless Sensor Networks Using Binary Data” Proceedings of the 38th Annual Conference on Information Sciences and Systems, Princeton, NJ, March 2004.
- [11] Xinbo Yu, Koushik Niyogi, Sharad Mehrotra and Nalini Venkatasubramanian “Adaptive Target Tracking in Sensor Networks” Communication Networks and Distributed Systems Modeling and Simulation conference (CNDS 04) , 2004 San Diego.
- [12] Jean-Paul Wagner and Razvan Cristescu “Power Control for Tracking in Sensor Networks” 2005 Conference On information Sciences and System, The Johns Hopkins university, March 16-18, 2005.
- [13] P.Bahl and V.N.Padmanabhan “RADAR: An in-building RF-based user location and tracking system” In Proceeding of the IEEE INFOCOM, pages 775-784, March 2000.
- [14] B.Hofman-Wellenhof, H.Lichtenegger, and J.Collins. “Global Positioning System: Theory and Practice” Springer Verlag, 4<sup>th</sup> ed, 1997.
- [15] D.Niculescu and B.Nath “Ad hoc positioning system (APS) using AoA” In Proceeding of IEEE INFOCOM 2003, pages 1734-1743, April 2003.
- [16] Alberto Cerpa, Deborah Estrin “ASCENT: Adaptive Self-Configuring Sensor Networks Topologies” IEEE transactions on Mobile Computing, vol.3, no.3, July-September 2004.

## Appendix

### A. Proof of Theorem 1 and 2

**Theorem 1.** Given a threshold  $H (0 \leq H \leq 1)$ ,  $\exists R \geq \sigma \sqrt{2 \ln \frac{1}{1-H}}$ , the probability that target will be in the round centered at prediction point  $(x_{pre-i}, y_{pre-i})$  with radius  $R$  at time  $i$  is greater than  $H$ .

**Proof:** Assume that  $P(x, y)$  is the probability that the target will be in the round centered at prediction point  $(x_{pre-i}, y_{pre-i})$  with radius  $R$  at time  $i$ , according to (1):

$$P(x, y) = \iint_{(x-x_{pre-i})^2+(y-y_{pre-i})^2 \leq R^2} f(x, y) dx dy . \text{ Given a threshold } H(0 \leq H \leq 1), \text{ if}$$

$$P(x, y) \geq H , \text{ which means that: } \iint_{(x-x_{pre-i})^2+(y-y_{pre-i})^2 \leq R^2} f(x, y) dx dy \geq H , \text{ let } u = \frac{x-x_{pre-i}}{\sigma} ,$$

$$v = \frac{y-y_{pre-i}}{\sigma}$$

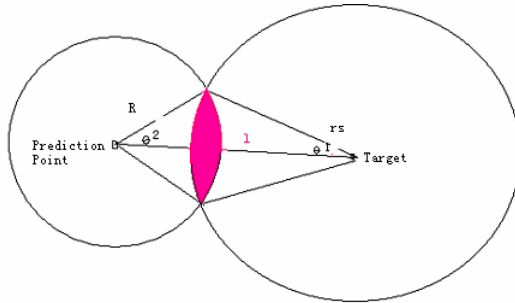
$$\Rightarrow \frac{1}{2\pi} \iint_{u^2+v^2 \leq r_0^2} e^{-\frac{u^2+v^2}{2}} dudv \geq H \Rightarrow r_0 \geq \sqrt{2 \ln \frac{1}{1-H}} \Rightarrow R \geq |\sigma| \sqrt{2 \ln \frac{1}{1-H}} \quad \square$$

**Theorem 2.** According to our pervious assumption: there are  $k$  sensors per  $\pi r_s^2$  averagely. ( $k \geq 1, k \in \mathbb{Z}^+$ ), if  $V_i \leq V_H$  , there will be at least one sensor inside Awakening Zone- $i$  that can detect target at time  $i$ .

**Proof:** According to Figure 2, if there is a sensor that can detect the target, it is inside the shadow  $S$ :

$$S = r_s^2 \theta_1 - r_s^2 \sin \theta_1 \cos \theta_1 + R^2 \theta_2 - R^2 \sin \theta_2 \cos \theta_2$$

$$\theta_1 = \cos^{-1} \left( \frac{r_s^2 + l^2 - R^2}{2rl} \right) \quad \theta_2 = \cos^{-1} \left( \frac{R^2 + l^2 - r_s^2}{2Rl} \right)$$



**Fig. 1.** An Example of AZ and the Target

Since there are  $k$  sensors per  $\pi r_s^2$  averagely, it is easy to know:  $S^*k / \pi r_s^2 \geq 1$  .

According to Definition 3,  $V_H$  is the result of equation (6) when  $S^*k / \pi r_s^2 = 1$  . So if

$V_i \leq V_H$  ,  $S^*k / \pi r_s^2 \geq 1$  which means that at least one sensor can detects the target.  $\square$

**B. Explanations of Variables in Paper**

**Table 1.** Explanations of variables appeared in the paper

$I_i(x_i, y_i)$	Target's position we detect at time i.	$r_s$	Detection range of sensors
k	Density of WSN	$\phi_a$	Power value of a sensor in active state
$\phi_s$	Power value of a sensor in sensing state	$\phi_i$	Power value of a sensor in idle state
$(x_{pre-i}, y_{pre-i})$	Prediction position of the target at time i	$V_{i-1}$	target's instantaneous speed at time i-1:
$\sigma$	The standard deviation	$H(0 \leq H \leq 1)$	threshold
$R = \sigma \sqrt{2 \ln \frac{1}{1-H}}$	The radius of AZ	$l_{max}$	$l_{max} = V_H * \Delta t$
$V_H$	The maximum speed t	d	Distance between the sensor and the target
$E_r$	Rest energy	$\alpha$	The percentage variable
F	decides whether the sensor would be queried by BS	$k_q(i)$	Number of sensors queried by BS at time i
$k(i)$	Number of sensors can detect the target at time i without TLS	$k^{\wedge}(i)$	Number of sensors can detect the target at time i with TLS
$\Delta t$	Time interval between time i-1 and i	$E_i$	Energy consumption between time i-1 and i without TLS
$E_i^{\wedge}$	Energy consumption between time i-1 and i with TLS	$\Delta E_i$	Difference between $E_i$ and $E_i^{\wedge}$

# Reducing Congestion in Real-Time Multi-party-tracking Sensor Network Applications

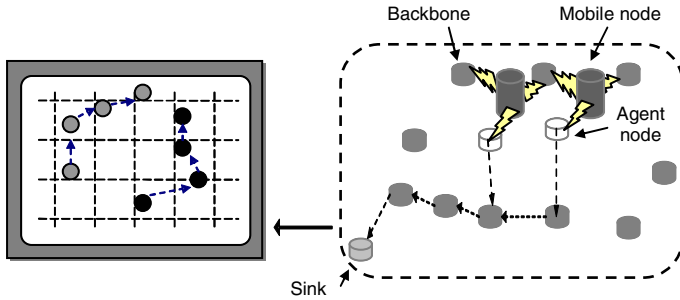
Wonwoo Jung, Sujeong Shin, Sukwon Choi, and Hojung Cha

Dept. of Computer Science, Yonsei University,  
134 Shinchon-dong, Seodaemun-gu, Seoul 120-749, Korea  
{wwjung, acid, sukwon, hjcha}@cs.yonsei.ac.kr

**Abstract.** This paper presents a framework for a congestion reduction mechanism in multiple objects tracking applications. The multiple objects tracking application has a real-time characteristic, through which all packets should be delivered in as timely fashion as possible. The proposed framework consists of sink nodes, backbone nodes, mobile nodes and agent nodes. The location information of the mobile node is delivered to the sink node using both localization and routing techniques, which consider the network congestion. The proposed congestion reduction technique avoids network congestion by decreasing the number of transmitting packets based on the idea that network congestion occurs when the total data in transit is greater than the total channel capacity. The decrement is achieved by both merging and attachment operations, which consolidate several small packets into one large packet in intermediate backbone nodes. The mechanism also considers a time-constrained delivery to assure the user's service request. The simulation is conducted using a TOSSIM simulator, and the result shows that the proposed mechanism improves performance in terms of the successful delivery ratio and the delay time.

## 1 Introduction

Wireless sensor network applications are largely classified into non-real-time applications and real-time applications. The non-real-time application transmits information gathered by sensor nodes in the sensor field to sink nodes without timing constraints by using multi-hop routing technologies. In this application, information is often overlapped, thus the amount of information is considered more important than the accuracy of information. Habitat monitoring is a representative application for a non-real-time application. The real-time application has a timing constraint: that is, the accuracy as well as the delivery time of information is important. Tracking mobile objects and monitoring the facility for disaster avoidance are good examples of real-time applications. Especially, tracking applications are considered the one of the major applications in sensor networks. In tracking applications, sensor nodes in the sensor fields track mobile objects, and transmit their location information to sink nodes. However, many of current approaches to track moving objects assume one or a few moving objects in the sensor field. As the sensor network becomes widely deployed, tracking applications should consider multiple objects for the practical reason. The application that tracks multiple mobile objects is called a Multi-Party Tracking application (MPT) in this paper.



**Fig. 1.** Proposed framework for MPT applications

In MPT, as the number of moving object increases, the amount of information also increases, and this results in network congestion. A hop-by-hop flow control mechanism such as CODA (Congestion Detection and Avoidance) [1] and fairness control mechanism [2] reduces the congestion occurrence by reducing the transmission rate locally [3]. However, the mechanism also induces transmission terminations or delay. A rate limiting mechanism such as ESRT (Event-to-Sink Reliable Transport) [4] adjusts the reporting rate in order to reduce network congestion based on the idea that network congestion occurs when the total data in transit is greater than the total channel capacity [5]. However, the mechanism assumes the overlap of information, and does not consider time-constrained delivery. Fusion [3] combines the hop-by-hop flow control mechanism and the rate limiting mechanism, and Kang et al [6] proposed the mechanism that distributes packets to neighbor nodes in order to reduce network congestion. This research for managing network congestion has focused on reducing congestion occurrence. However, they overlooked both the user's service requirement and the time-constrained packet delivery. Moreover, this research is not suitable for MPT applications due to their dynamic adjustment of the event generation rate, because the MPT application expects to receive information with a fixed rate within a certain threshold.

This paper proposes a framework and a congestion reduction mechanism, called M&A (Merge and Attach), for the MPT application. The mechanism adjusts the packet generation rate, and reduces the number of network packets by consolidating several small packets into one large packet at intermediate backbone nodes. It also considers a time constrained delivery, thus merging and attachment operations occur selectively based on real-time constraints. The paper is organized as follows. Section 2 describes the proposed framework and the congestion reduction technique. Section 3 discusses the experiment's results, and Section 4 concludes the paper.

## 2 Congestion Reduction Mechanism for Multi-party-tracking

The MPT application monitors the current location of various numbers of mobile nodes in sensor networks and transmits them to the sink node. Figure 1 shows the proposed framework of the MPT system. Since the MPT application is a real-time application, every information for location must arrive at the sink node within a certain deadline. In other words, location information must be transmitted immediately and the delay time occurring in the relaying process should be minimized. Other char-



acteristics of MPT are a fixed reporting rate and single datum per single event, which means no redundant location information.

The proposed system consists of four types of sensor nodes. Mobile nodes, which calculate their location periodically using a localization algorithm, are attached to moving objects. Backbone nodes are aware of their global positions and are statically placed. When they receive beacon messages from mobile nodes, they calculate the distance between backbone nodes and mobile nodes, and relay the location information of mobile nodes to the sink node. Agent nodes, which are chosen by mobile objects, are backbone nodes, and relay the location information of mobile nodes to the sink node. Finally, the sink node is connected to the main system, and has an unlimited power source. The distance between the mobile node and the backbone node is calculated by the method used in the Cricket System [7], and the agent node is selected by the method used in SPEED [8]. The working scenario of MPT is followed. When multiple mobile nodes move, they broadcast beacon messages via RF and ultra sound periodically. Then, backbone nodes deployed in the sensor field receive the beacon message and calculate the distance between the backbone node and the mobile node. The calculated distance is transmitted to the mobile node, and then the mobile node selects three backbone nodes among them and calculates its location using the distance between the mobile node and the selected three backbone nodes. The location information is transmitted to the sink node through the agent node that is chosen from among three backbone nodes previously selected using multi-hop routing techniques. During the data transmission, the congestion reduction mechanism is conducted in order to reduce network traffic. The proposed system operates in two phases: the localization phase and the transport phase.

## 2.1 Localization Phase

Sampling rate  $R_s$ , which indicates the amount of transmitted data between a mobile node and a backbone node per unit time, is controlled by the mobile node according to the velocity change.  $R_s$  is equal to or less than the reporting rate  $R_r$ , which is the amount of received data from a user per unit time, and decided by the user. The user receives the location information of the mobile node regardless of whether the data has actually arrived at the sink node or not. At the beginning of the tracking process, the mobile node calculates its own location with the initial sampling rate  $R_{init}$ . The velocity is a vector value consisting of speed  $v$  and direction  $d$ ; therefore the velocity change is determined by both the speed change  $\Delta v$  and the direction change  $\Delta d$ . Figure 2 shows the reconfigure process of  $R_s$  depending on the velocity change of the mobile node.

When  $\Delta v$  is larger than the speed threshold  $\gamma_{\Delta v}$  (Case 1),  $R_s$  is set to  $R_{init}$ . When  $\Delta d$  is larger than the direction threshold  $\gamma_{\Delta d}$  (Case 2),  $R_s$  is set to  $R_{init}$ . In other cases (Case 3),  $R_s$  is set to the default minimum value  $R_m$ . When the mobile node calculates its location, it decides whether the location information should be transmitted to the sink node or not, because the sink node can predict its location when the speed of the mobile node is moderate. Equation 1 and Equation 2 show the condition of the decision.

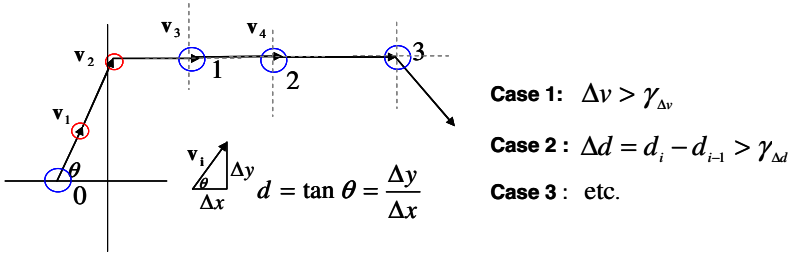


Fig. 2. Reconfiguration of the sampling rate

$$G(\mathbf{v}_{i-1}, \mathbf{v}_i) = \begin{cases} 0 & \text{if } \mathbf{v}_i = \mathbf{v}_{i-1} \\ 1 & \text{otherwise} \end{cases} \quad (1)$$

$$F(G(\mathbf{v}_{i-1}, \mathbf{v}_i), \Delta t) = \begin{cases} SEND & \text{if } G(\mathbf{v}_{i-1}, \mathbf{v}_i) = 1 \text{ or } \Delta t \geq R_m \\ NOT SEND & \text{otherwise} \end{cases} \quad (2)$$

When the mobile node decides to send the location information to the sink node, it selects one agent node, which has the biggest value of *speed* based on the SPEED [8] algorithm, from the backbone nodes (Equation 3). In Equation 3, *Hop Delay* indicates the elapsed time during the location information transmitted from the mobile node to the backbone node, and *distance\_mobile* is the distance between the sink node and the mobile node. The distance between the sink node and the backbone node is denoted as *distance\_backbone*. When the mobile node selects the agent node, it considers the distance as well as the communication ability of backbone nodes.

$$speed = \frac{distance\_mobile - distance\_backbone}{Hop\ Delay} \quad (3)$$

## 2.2 Transport Phase

When the agent node receives the location information of the mobile node, the second phase begins. Since MPT requires only 7 bytes for transmitting the location information of mobile nodes to the sink node in this phase, it is possible to consolidate several packets into one packet in order to reduce the network traffic because MICA2, operating on TinyOS, can transmit a packet with a size of 36bytes at the maximum [9]. Figure 3 shows the structure of the backbone node, which handles the merging and attachment operation (M&A), and relays packets to the sink node. The merging operation consolidates several packets that are generated at a similar time, and the attachment operation suspends and attaches packets considering their delay time. The merging and attachment operation are applied separately based on two values: the elapsed time  $\Sigma\Delta t$ , which means the duration between the current time and the packet creation time, and the threshold  $\tau$ . If the elapsed time is smaller than the threshold, which means the data is valuable in terms of the real-time concept, the merging operation is applied. Otherwise, the attachment operation is applied in order to give more precise location information for mobile nodes to the user; however, the data is out-of-date.

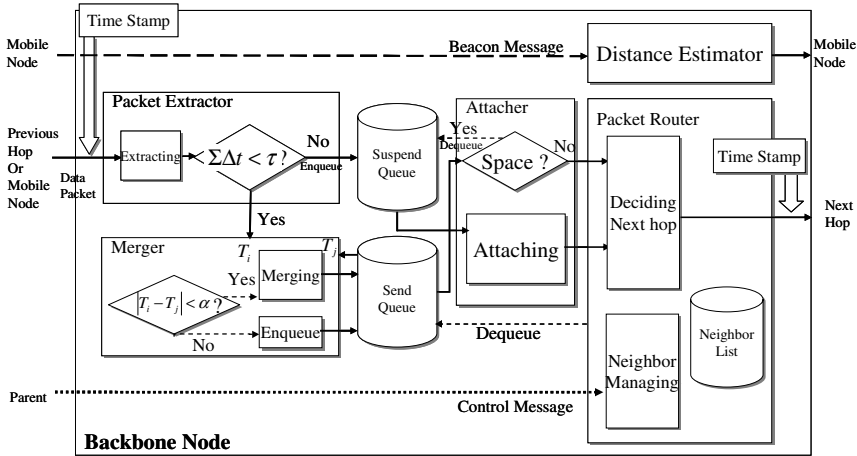


Fig. 3. Structure of the backbone node

The Packet Extractor compares  $\Sigma\Delta t$  and  $\tau$ , and determines which operation should be applied. If the merging operation is selected, the elapsed time of packets, which are stored at the send queue, are compared with one another. If the difference between them is within the range  $\alpha$ , the Packet Extractor examines the size of the packets and merges them if it is possible. When the difference exceeds  $\alpha$ , the merging operation is not applied. If the attachment operation is selected, the packet currently received does not have to be transmitted as soon as possible, because it cannot be delivered to the sink node on time; therefore it is stored at the suspend queue. These suspended packets are attached only if they can be attached to the packet that is going to transmit to the next node. Although attached packets cannot be delivered within a time constraint, they can give the user a hint about the moving track of the mobile node in the past.

### 3 Evaluation

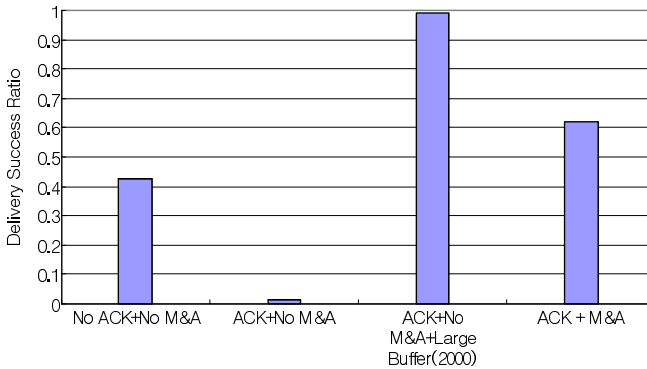
Simulations were conducted with a TOSSIM [10] simulator of TinyOS, which is designed for MICA hardware. It simulates CC1000 [11] used at MICA2 selectively, and all simulations in the paper were conducted using CC1000 for a radio module. Currently, TOSSIM based on B-MAC [12] cannot use ACK messages. Kang et al. [13] showed that an ACK-based retransmission policy is required to fill a transmission buffer. Therefore, we modified the simulator to use ACK messages selectively. Table 1 shows simulation parameters and values used in the simulation.

With this setup, each backbone node has 4 direct-communicable neighbor nodes at its top, bottom, left, and right directions, because CC1000 has about a 15m radio range. To emulate mobile nodes, some backbone nodes produce traffic. We adopted a simple routing policy, which selects the closest neighbor node to the sink node randomly. Based on the preliminary experiment, the maximum allowed time  $\tau$ , which is used to examine whether the real-time constraints of incoming packet at the sink node is satisfied, is set to 1000ms, which is the same as the reporting period. Because  $\tau$ ,

which is larger than the reporting period, decreases the delivery success ratio due to the increase of packet losses caused by send queue overflows. The merging boundary time  $\alpha$ , which is used to examine whether a new packet should be merged with other packets in the send queue or not, is set to 300ms, because small  $\alpha$  decreases the delivery success ratio.

**Table 1.** Simulation parameters and values

Parameter	Value
Distance between backbone nodes	12 (Meter)
Field size	108×108 (m <sup>2</sup> )
Number of backbone nodes	10×10
Number of simulations in each	5
Simulation time	5 (Minutes)
Sink location	Center / Corner
Number of mobile nodes	30
Arrangement of mobile nodes	Random / Dense deployed
Size of send queue	8×36 (Bytes)
Size of suspend queue	32×7 (Bytes)



**Fig. 4.** Delivery success ratio in various settings

Figure 4 represents the delivery success ratio with 4 different cases according to the combination of two policies: ACK-based retransmission policy and M&A policy. When the retransmission policy is not applied, the delivery success ratio is about 42.5%, regardless of whether it is applying the M&A policy or not. Since the M&A policy utilizes packets in the transmission buffer, the retransmission policy should be applied to adopt the M&A policy successfully. When the retransmission policy is applied without the M&A policy, the delivery success ratio is about 1.5%, which is the worst case of all. The worst delivery success ratio is due to the heavy traffic caused by retransmissions, which causes the transmission buffer to overflow. When the size of the transmission buffer is increased from 8×36bytes to 2000×36bytes, the

delivery success ratio is about 100%, and it is the best case of all. Although the large buffer improves the delivery success ratio, the elapsed time of almost every packet in this case is more than 35,000ms. Therefore, it is not applicable to MPT applications, which have a real-time characteristic. When the retransmission and M&A policy are applied together, it achieves about a 62.0% delivery success ratio. Consequently, adopting both the ACK-based retransmission policy and M&A policy achieve the best performance with the consideration of the real-time constraint.

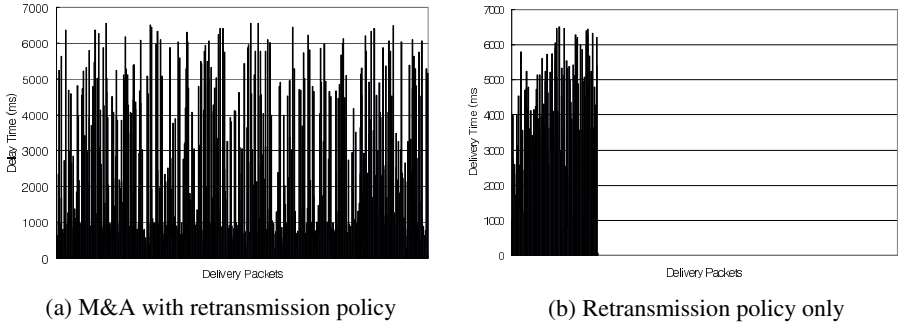
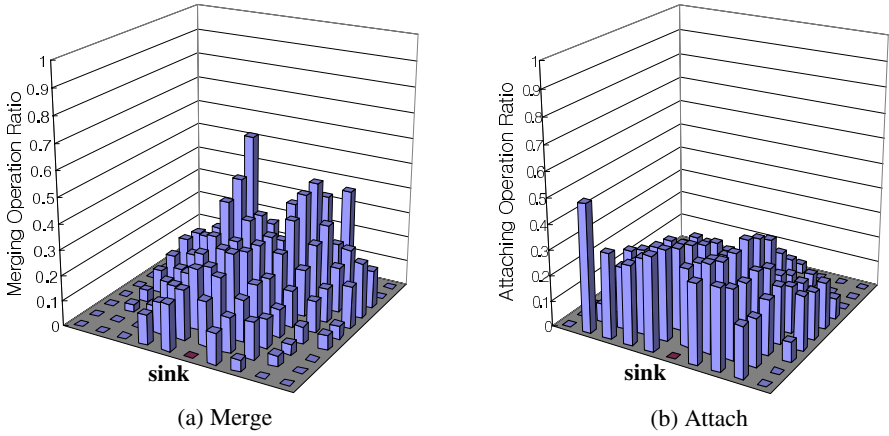


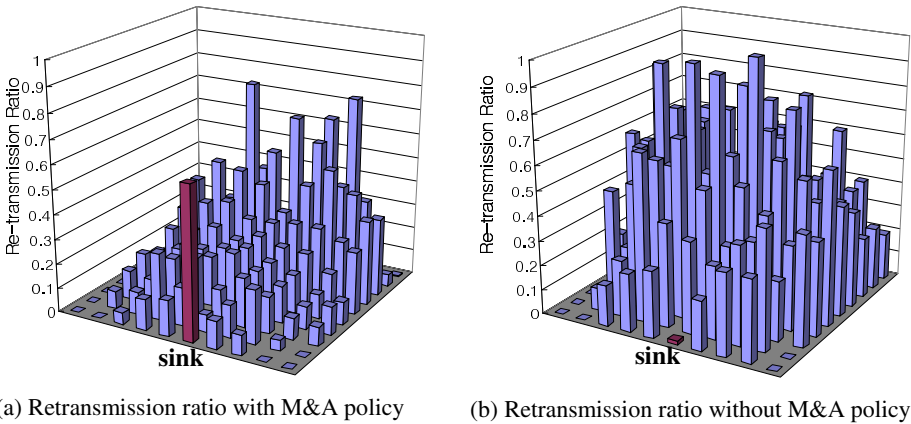
Fig. 5. Delay time distribution

Figure 5 shows the delay times of the first 1000 packets. Figure 5(a) represents the delay time when both the retransmission and M&A policy are applied. In this case, about 60% of packets arrived at the sink node within the maximum allowed time  $\tau$ , 1 second. When the retransmission policy is applied without the M&A policy, the delivery success ratio of this case is very low, and the delay time of arrived packets is higher than the other cases, as shown in Figure 5(b). When neither the retransmission nor the M&A policy is applied, all packets successfully delivered to the sink node arrived within the allowed time. However, in this case, the delivery success ratio is about 40% lower than Figure 5(a). Therefore, applying both the M&A policy and the retransmission policy achieve the best performance when considering both the delay time and the delivery success ratio.

Figure 6 shows the number of merging and attachment operations in the backbone nodes, when both the retransmission and M&A policy are applied. The Z-axis shows the ratio of both merging and attachment operations. The position of each bar in Figure 6 is the same as the position of each node in the simulation. When nodes are located near to the event source, the merging operation occurs more frequently. In addition, the number of merging operations decreases, as nodes are located near to the sink node. On the contrary, the attachment operation occurs more frequently when nodes are located near to the sink node or if they are located far from the event source as Figure 6(b) shows. This is because the packet that has a long elapsed time has a high probability to attach. The high attachment operation ratio near the sink node reduces the traffic and the number of packets in the send queue. Therefore, the ratio of merging operating in this case decreases, because the attachment operation ratio is higher than the merging operation ratio near the sink node.



**Fig. 6.** Merging and attachment operation ratio



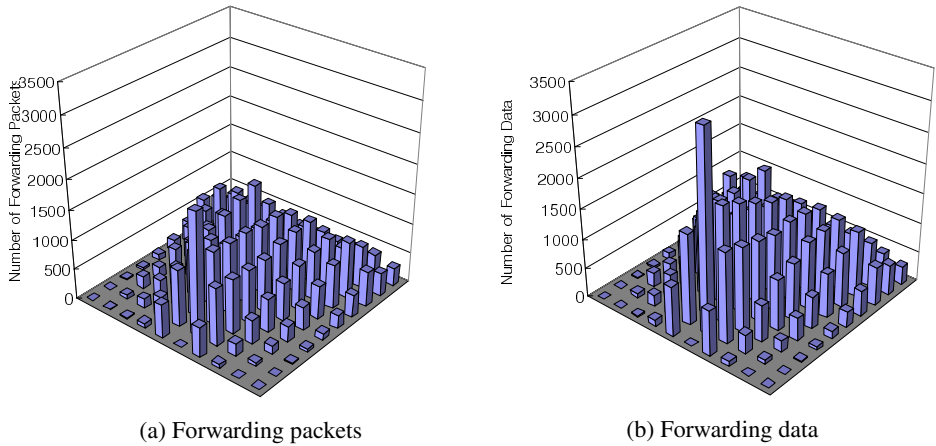
**Fig. 7.** Retransmission ratio whether applying M&A policy

Figure 7 shows the retransmission ratio whether applying M&A policy or not. The Z-axis in Figure 7 shows the retransmission ratio for one transmission at a node. In addition, the bar of the sink node represents the transmission success ratio. When the M&A policy is not applied, as shown in Figure 7(b), the retransmission ratio increases when nodes are located near to the event source; it decreases as nodes are located nearer to the sink node. This is because the packet loss caused by the congestion reduces the network traffic, as nodes are located near to the sink node. When the M&A policy is applied, as shown in Figure 7(a), the retransmission ratio decreases continuously, as nodes are located near to the sink node. It proves that the merging and attachment operations reduce network congestion occurrences.

Figure 8 shows the total number of transmitted packets and the amount of transmitted data. The total number of transmitted packets decreases as packets are transmitted

toward the sink node. On the other hand, the total amount of transmitted data increases as packets are transmitted toward the sink node. When the M&A policy is applied, the transmitted packets are reduced by 26% of the total amount of data. It proves that the merging and attachment operation reduces the total number of packets even though the amount of data is increased.

As a result, the proposed M&A policy shows the improvement of the delivery success ratio, when it is applied with the retransmission policy. Although performance in terms of the delay time may be worse than when it is applied without the retransmission policy, the degradation is trivial. Moreover, from the maximum allowed time of view, it achieves a good performance.



**Fig. 8.** Number of forwarding packets and amount of forwarding data

## 5 Conclusions

There are a number of approaches to solving the congestion problem in sensor networks; however, they cannot be applied to MPT application because of its real-time constraints. M&A uses a merging or attachment operation in order to decrease the number of packets in networks. Because the M&A mechanism does not suppress the data generation rate and does not postpone the forward of packets, it is suitable for real-time applications. The simulation result shows that using the M&A mechanism improves the delivery success ratio more than any other cases. The delay time is also examined in order to evaluate real-time constraints. When using the M&A mechanism with the retransmission mechanism the delay time is a little longer than without using the retransmission mechanism. However, it does not mean that the M&A mechanism with the retransmission mechanism performs badly in terms of real-time constraints, since packets that arrived at the sink node within  $\tau$  are considered as packets that satisfy real-time constraints in MPT applications. From this point of view, we can conclude that the case with M&A and the retransmission mechanism showed similar performance to the case without the retransmission mechanism.

We are implementing the proposed framework and M&A mechanism to real hardware based on MSP430 MPU. Our final goal is to implement the MPT application on real hardware, including the congestion mechanism and localization algorithms for tracking itself.

## Acknowledgements

This work was supported by the National Research Laboratory (NRL) program of the Korean Science and Engineering Foundation (2005-01352) and the ITRC Program (MMRC) of IITA, Korea.

## References

1. C.-Y. Wan, S. B. Eisenman, and A. T. Campbell.: CODA: Congestion Detection and Avoidance in Sensor Networks. In Proc. of ACM SenSys 2003, Los Angeles, USA, (2003)
2. C.-T. Ee and R. Bajcsy.: Congestion Control and Fairness for Many-to-One Routing in Sensor Networks. In Proc. ACM SenSys 2004, Baltimore, MD, (2004)
3. B. Hull, K. Jamieson, and H. Balakrishnan.: Mitigating Congestion in Wireless Sensor Networks. In Proc. of ACM SenSys 2004, Baltimore, MD (2004)
4. Y. Sankarasubramaniam, O. B. Akan, and I. F. Akyildiz.: ESRT: Event-to-Sink Reliable Transport in Wireless Sensor Networks. In Proc. of MobiHoc 2003, Annapolis, Maryland (2003)
5. S. Tilak, N. B. Abu-Ghazaleh, and W. Heinzelman.: Infrastructure Tradeoffs for Sensor Networks. In Proc. of WSNA 2002, Atlanta (2002), 49–58
6. J. Kang, Y. Zhang, B. Nath, and S. Yu.: Adaptive Resource Control Scheme to Alleviate Congestion Control in Sensor Networks. In Proc. of the 1st Workshop on Broadcast Advanced Sensor Networks (BASENETS), San Jose, CA (2004)
7. N. B. Priyantha, A. Chakraborty, and H. Balakrishnan.: The Cricket Location-Support System, In Proc. of 6th ACM MOBICOM, Boston, MA (2000)
8. T. He, J. A. Stankovic, C. Lu, and T. F. Abdelzaher.: SPEED: A Stateless Protocol for Real-Time Communication in Sensor Networks. In Proc. of International Conference on Distributed Computing Systems (ICDCS 2003), Providence, RI (2003)
9. <http://www.tinyos.net>
10. P. Levis, N. Lee, M. Welsh, and D. Culler.: Tossim: Accurate and Scalable Simulation of Entire Tinyos Applications, In Proc. of ACM SenSys 2003, Los Angeles, USA (2003)
11. [http://www.chipcon.com/files/CC1000\\_Data\\_Sheet\\_2\\_2.pdf](http://www.chipcon.com/files/CC1000_Data_Sheet_2_2.pdf)
12. J. Polastre, J. Hill, and D. Culler.: Versatile Low Power Media Access for Wireless Sensor Networks, In Proc. ACM SenSys 2004, Baltimore, MD (2004)
13. J. Kang, Y. Zhang, and B. Nath.: Accurate and Energy-efficient Congestion Level Measurement in Ad Hoc Networks, In Proc. of the IEEE Wireless Communications and Networking Conference (WCNC), New Orleans, LA (2005)



# Variable-Radii Method Based on Probing Mechanism (VRPM): An Energy Conservation Method for Wireless Active Sensor Networks

Qi Zhou, Takuya Asaka, and Tatsuro Takahashi

Department of Communications and Computer Engineering,  
Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan  
zhou@cube.kuee.kyoto-u.ac.jp  
{asaka, ttakahashi}@i.kyoto-u.ac.jp

**Abstract.** Wireless sensor networks, which are composed of advanced MEMS (Micro-Electro-Mechanical-Systems) called sensor nodes, has been broadly researched as ad-hoc networks recently. Since sensor nodes are powered by limited-life batteries and are usually deployed in severe environments, they need to be able to conserve their power consumption as well as guarantee broad coverage and connectivity. In active sensor networks, such as RFID sensing systems, shorter sensing and communication ranges will result in a higher measuring precision and more efficient energy dissipation. In this paper, we propose a novel self-controlling method for sensor nodes to decide the most appropriate sensing radius, communication radius, and an active/sleep schedule simultaneously. The energy-saving effect is proved in our simulation section.

## 1 Introduction

A wireless sensor network is an ad-hoc network disposed to observe a certain environment, sense useful information such as temperature, humidity, pressure, light, voice, and send data efficiently to the sensor network's gateway called the BS (Base Station). The BS is connected to an IP network, such as a LAN or the internet. Users can post requirements to the sensor network base station and obtain the desired information from it.

A sensor node, which is typically composed of sensor, A/D converter, processor, memory, RF, and battery modules, is the element unit of sensor network employed to measure the surroundings. Sensor nodes usually operate in severe conditions, meaning that repair and battery exchange are difficult to do. For these special features as a network, an energy-efficient, self-scheduling method is strongly required to conserve energy by reducing energy redundancy.

There are two major types of sensors. Active sensor systems interact with the environment and observe how their actions affect the environment. Examples of active systems include RF transmitters such as RFID tag reader/writers, IR transmitters, structured lighting, and most sonar and radars. Passive sensor systems sense ambient radiation or signals. Passive systems include GPS, ambient audio, and passive motion detectors [1].

This study focuses on a ubiquitous RFID reader/writers networking, a typical model of active sensor networks. Countless objects with RFID tags attached are scattered across an enormous domain, and in order to identify these tags by RFID technology, large quantities of RFID reader/writers are introduced to construct a large-scale sensor network. In an RFID reader/writers network, RFID reader/writers act as sensor nodes to collect tags' information using an RFID sensing system energy model, and communicate with other reader/writers or base stations to propagate data with an RF communication energy model. We will investigate the characteristic of active sensor networks to find out the dominant energy consumption fraction, and try to propose an optimal energy-aware scheduling method for active sensor networks, by varying sensing and communication radii.

The remainder of the paper is organized as follows. In the next section, we will review related work on the coverage and energy-efficiency problem briefly. Then, on the energy model, sensor nodes will be divided into several essential modules and reasonable analytic models will be abstracted in Section 3. Based on energy dissipation models, we will introduce VRPM and explain its elemental steps in Section 4. In Section 5 we use simulations to comprehensively evaluate the system's parameters and energy needs, which proves the efficiency of our proposal. Section 6 presents our conclusions and future work.

## 2 Related Work

Di Tian and colleagues proposed a solution that requires every node to know all its neighbors' positions before its schedule is determined [6]. This feature guarantees a 100% sensing coverage, but incurs a heavy communication overhead led by a priori knowledge about all neighbors. Also, this rule underestimates the area that the neighbor nodes can cover, which leads to excess energy consumption.

Fan Ye and colleagues proposed a simple localized protocol named PEAS for dynamically adjusting the schedule with a probing mechanism [5]. In this solution, after sleeping for time  $T_s$ , node wakes up and broadcasts a probing message within a certain probing range  $R_p$  and waits for a reply. If no reply is received before a certain interval elapses, the node will transfer its status to active and remain active until the end of its battery lifetime. Although PEAS is a probabilistic method, which means full area coverage cannot be ensured, we have testified that the coverage of a monitored area is close to 1 if the threshold probing distance is less than the sensing radius.

Most recent studies on the coverage problem of sensor networks, like PEAS, assume that the sensing radius of each node is of one size. However, Fig. 1 shows the situation of covering a small domain, which implies that fixed sensing and communication radii will result in a severe redundancy problem. The redundancy we mentioned here may comprise of the following three points:

- (1) It is redundant to sense the same point with multiple sensor nodes simultaneously.
- (2) It is redundant to send the same sensed data multiple times due to (1).
- (3) It is redundant for one node to transmit its sensed data to its neighbors within at least  $2R_s$ , which may be unnecessary.

Variable-radii algorithms have been proposed recently to resolve redundancy problem caused by radius fixation. In fact, it is often assumed that a wireless device can change its transmission range to save energy [10], [11].

Jie Wu and Shuhui Yang [8] introduced a method based on one of three models. In model I, the sensing radius is fixed as maximum radius  $R$ . Model II adopts two sensing radii. Ideally, the count proportion of large disks to small disks is 1:2 (Fig. 2(a)). In model III, three sensing radii can be chosen by each sensor node and the count proportion here is 1:6:2 (Fig. 2(b)). Models II and III can be regarded as adopting a half-variable-radii method, which involves several discrete radii from 0 to  $R$ . However, due to the inequality of radius distribution (Fig. 8(a)), sensor nodes with large radii will exhaust their power much faster than those with small radii. Moreover, if there are a large proportion of small-radius sensor nodes, the involvement of too many sensor nodes will also result in energy redundancy.

Zongheng Zhou and colleagues [9] developed a series of algorithms to settle the variable-radii connected sensor coverage (VRCSC) problem. Since their goal is to select a subset of sensors with a specified sensing radius so that each point in the query region can be sensed by at least one of the selected sensors, each sensor node should know its neighbors' coordinates precisely, and a complicated calculation process has to be followed a distributed manner.

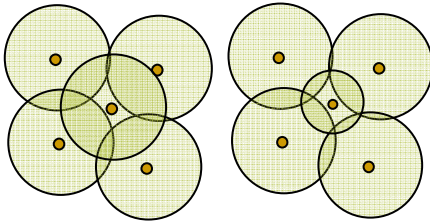
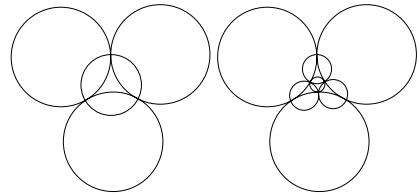


Fig. 1. Case of covering a small domain



(a) Model II (b) Model III

Fig. 2. Discrete variable-radii model

### 3 Energy Model for Active Sensor Networks

To discuss energy efficiency, we first need to consider the energy dissipation model. In an active sensor network system, the dominant energy consumption units should be the sensing module, MCU (Micro Controlling Unit) and the wireless communication module.

As for communication module, here we consider a simple transceiver model [2]. Suppose  $E_{elec}$  represents the energy of bits dissipated from the sensor in order to activate the transceiver circuitry that is present in both the receive and transmit modes, and  $\epsilon_{amp}$  represents the energy of bits dissipated from the transmit amplifier.

In this simple radio-frequency model, to transmit  $k$  bits to a distance  $d$  meters assuming only path loss, the energy consumption could be:

$$E_{Tx}(k, d) = E_{elec} * k + \epsilon_{amp} * k * d^\beta. \tag{1}$$

To receive  $k$  bits, the model of energy consumption could be:

$$E_{Rx}(k) = E_{elec} * k. \quad (2)$$

In a typical wireless communication model,  $\beta$  can be set to 2.

Klaus Finkenzeller [3] and Zheng Zhu [4] introduced the model of electromagnetic radiation waves energy used in RFID reader/writers systems. As we know, some of the power will be absorbed by the targets (RFID tags), while other fractions will be scattered into different directions with different amplitude. Finally, a very small part of the power will be reflected back to the RFID reader/writers. For simplicity, we are investigating an ideal model assuming that RFID reader/writers send RF signals without any specific direction and that RF energy is scattered evenly in gradual concentric circles. We also assume that the RF energy reflected simultaneously by all tags lying in the RF reader's sensing domain, and that energy is reflected completely without any loss or absorption. Thus, for minimum energy considerations, if one RFID reader/writer scatters  $\epsilon_{amp}kd^\beta$  energy, the return energy from the farthest tag it can reach is supposed to be  $\epsilon_{amp}kd^\beta(1/d^{2\beta})$ , which indicates that the reader/writer's receive power is inversely proportional to the power of four (if  $\beta=2$ ) of the distance between the tag and the reader/writer. In other words, if we double the distance, the received power at the reader/writer would be decreased to 1/16 of the previous power.

Based on the discussion above, Eq. (1) can be modified to adapt the RFID reader/writer model as:

$$E_{Tx}(k, d) = E_{elec} * k + \epsilon_{amp} * k * d^{2\beta}. \quad (3)$$

## 4 Variable-Radii Method Based on Probing Mechanism (VRPM)

Our purpose is to propose a simple method that can be applied to distributed sensor nodes easily, without knowledge of neighbors' precise positions, while maintaining the radii distribution balance simultaneously. Although we do not try to ensure that each point of the monitored region can be covered, we will show in Section 5 that a high probabilistic coverage (very close to 1) can be provided.

In order to conserve energy, sensor nodes are self-configured to one of three statuses: Sleep, Active, and Probing. We propose a method for each sensor node to decide its sensing and communication radii, and Active/Sleep schedule as well. Once sensing/communication radii are decided, they will remain the same throughout the node's lifetime unless it is necessary to decide radii and schedule once more. Here are the essential steps for our VRPM method, supposing the current sensor node is named  $X$ . We also define  $R$  and  $2R$  as maximum sensing radius and maximum communication radius respectively.

**Step 1:** Initially, all the sensor nodes begin with the status of sleep.

**Step 2:** After an exponentially distributed random time  $T_s$ , node  $X$  wakes up and broadcasts a probing message within the maximum probing range (maximum communication range)  $2R$  and waits for a reply to determine active nodes.

- (a) If no reply is received within a certain interval, it will transfer its status to active. Sensing radius  $R_s$  is determined randomly in the range of  $[0, R]$ , while the communication radius is fixed as  $2R$ .

(b) If a reply is received from a certain node  $A$ , node  $X$  will calculate the distance between  $A$  and itself (denoted as  $Dis(A, X)$ ). If  $X$  is covered by  $A$ , that is  $Dis(A, X) \leq R_s(A)$ ,  $X$  will go back to sleep. If not, that is  $Dis(A, X) > R_s(A)$ ,  $X$  will switch to active with a sensing radius of

$$\alpha \times \frac{1}{n} \sum_{i=1}^n (Dis(A_i, X) - R_s(A_i)), \tag{4}$$

and with a communication radius of

$$Max(Dis(A_i, X)), \tag{5}$$

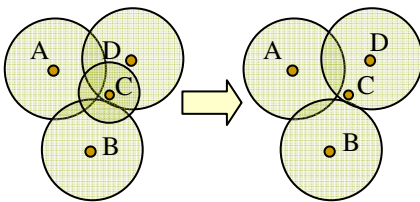
where  $n$  represents the count of sensor nodes on duty within  $X$ 's probing range, and  $\alpha$  is an overlap coefficient.  $Max(Dis(A_i, X))$  represents the maximum of all values of  $Dis(A_i, X)$ .

**Step 3:** Once node  $X$  transfers its status from sleep to active, it will remain active until the end of its battery lifetime. When  $X$  is exhausted, it just dies silently and the lack of coverage will be compensated for by other sensor nodes when their next  $T_s$  comes.

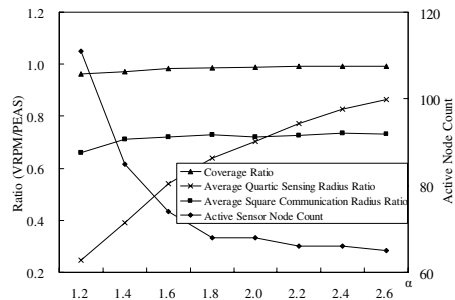
**Step 4:** If node  $X$  finds it comes to cover another active node when it tries to transfer from sleep to active status, it should instruct the covered one to sleep.

Steps 1-3 explain a basic method for each sensor node to decide when to transfer its status from sleep to active, and what the appropriate sensing and communication radii are that it should follow after its status transfer. In sensing the radius decision,  $Dis(A, X) - R_s(A)$  implies the minimum necessary radius to every single active neighbor. After the mean value is calculated, the sensing radius may still not be large enough to achieve the wide coverage desired, so overlap coefficient  $\alpha$  is introduced to provide an adaptive modification. A performance evaluation for  $\alpha$  will be discussed in the simulation section (Section 5).

Also, Step 4 resolves a serious problem that occurs because of the radius variation. Fig. 3 illustrates a situation supposing a radius determination sequence is nodes  $A, B, C$ , and  $D$ . A different consequence will arise if the determination sequence is nodes  $A, B, D$ , and  $C$ . It is believed that the latter one can reduce coverage redundancy, and it fits our original design better.



**Fig. 3.** Node  $D$  instructs  $C$  to sleep, Step (4) of VRPM



**Fig. 4.** Evaluation of the overlap coefficient  $\alpha$

To ensure the connectivity of the sensor nodes, we suppose that every sensor node should have the ability to communicate with all its reachable neighbors, without considering any routing protocols. Under this rule,  $Max(Dis(A_i, X))$  is introduced to determine communication radius, which may not be the smallest efficient value. A more precise communication radius control algorithm will be studied as our future work.

## 5 Performance Evaluation and Simulation

We have programmed a simulator to prove our proposal efficient by comparing it with the conventional PEAS method. All the sensor nodes are deployed randomly to a domain measuring  $100m \times 100m$ . Every sensor node judges whether it is necessary to be on duty, and determines its appropriate sensing/communication radii with the VRPM method. We adopt a simple model that every sensor node sense  $k$  bit ambient data every 1 second but only sends them to one of its neighbors every 100 seconds. That is to say, 101 seconds can be regarded as one cycle, in which every sensor node accomplishes 100 times sensing and once data propagation.

When probing,  $kp$  bit is assumed to be sent. As the  $kp$  is so small that communication energy dissipation can be ignored in probing status. Moreover, as the probing interval  $T_s$  is an exponentially distributed random time while sensor operates every one second, all the energy dissipation in probing status can be ignored if we observe for a long time enough.

**Table 1.** Simulation parameters

Parameter	Sensing Unit	Communication Unit
Maximum Radius	10 m	20 m
$\beta$	4	2
$E_{elec}$	50 nJ/bit	50 nJ/bit
$\epsilon_{amp}$	100 pJ/bit/m <sup>4</sup>	100 pJ/bit/m <sup>2</sup>

**Table 2.** Parameter  $\beta$  settings

	$\beta_s$	$\beta_c$
(a)	2	2
(b)	2	4
(c)	4	2
(d)	4	4

**Table 3.** Energy consumption (unit: mW)

Status	MCU[7]	Sensing Unit	Communication Unit
Sleep	0	0	0
Probing	16.5	0	$(0.05+10^{-4}Rc^2)k_p$
Active (Sensing)	16.5	$(0.05+10^{-4}Rs^4)k$	0
Active (Communication)	16.5	0	$(0.05+10^{-4}Rc^2)100k+0.05*100k$

### 5.1 Overlap Coefficient $\alpha$

Figure 4 shows the variations of Coverage ratio ( $Coverage_{VRPM}/Coverage_{PEAS}$ ), Average quartic sensing radius ratio ( $Rs^4/R^4$ ), Average square communication radius ratio ( $Rc^2/(2R)^2$ ), and Active sensor node count at an overlap coefficient  $\alpha$  range of (1.2, 2.6). As we can confirm from the chart,  $\alpha$  essentially implies the criterion of  $Rs^4/R^4$ , and VRPM provides a high coverage even if  $\alpha$  is much less than 1. Also,  $Rc^2/(2R)^2$  is a stable value because the maximum distance is adopted as the communication radius

in our method. However, the point we should concentrate on is the trade-off between the active sensor node count and the average sensing radius.

## 5.2 Energy Dissipation

Table 1 shows the reasonable parameters used in our simulation for the energy dissipation model, and the energetic consumption volume can be formulated according to Table 3, where  $R_c$  and  $R_s$  represent the communication radius and sensing radius, respectively.

As we have mentioned in Section 3,  $\beta_s = 4$  and  $\beta_c = 2$  are regarded as a typical parameter set in an active sensor network, according to Table 3, one cycle's total power  $P_{total}$  in the entire sensor network can be calculated with the following equation:

$$\begin{aligned} P_{total} &= (P_{sensing} + P_{communication}) * N_{Active} \\ &= \{ [16.5 + (0.05 + 10^{-4} R_s^4) k] * 100 \} + \{ [16.5 + (0.05 + 10^{-4} R_c^2) 100k] \\ &\quad + (16.5 + 0.05 * 100k) \} * N_{Active}, \end{aligned} \quad (6)$$

where  $P_{sensing}$  and  $P_{communication}$  denote power consumption by sensing and communication modules respectively, and  $N_{Active}$  represents the active sensor node count. The energy dissipation due to probing mechanism can be ignored approximately because the probing traffic volumes  $k_p$  is much less than the sensing traffic volumes  $k$ . In PEAS,  $R_s = 10$ ,  $R_c = 20$ ,  $N_{Active} = 67.2$  and

$$P_{total} = (1683 + 119k) * 67.2. \quad (7)$$

In the same way, we calculated all formulas by the variation of  $\alpha$  and produced the graphs in Fig. 5(c).

The results indicate that VRPM performs well in almost all traffic volumes  $k$ . In particular, as the traffic grows, VRPM even manages to reduce the energy consumption by about 40% in comparison to PEAS. As another consequence, we confirmed that a too small overlap coefficient does not bring us high performance because too many sensor nodes are involved in the sensing service. On the other hand, a large overlap coefficient implies a large average sensing radius, which results in a performance close to PEAS.

For universality of the sensing and communication energy model, we varied the essential parameter  $\beta$  of energy dissipation model as in Table 2, where we define  $\beta_s$  and  $\beta_c$  as the exponent for sensing and communication respectively, and the energy dissipation results are listed in Fig. 5 (a) to (d), where the vertical axis is in the unit of the power ratio dividing VRPM by PEAS.

Since the communication radius is much larger than the sensing radius (approximately double), the dominating partition of energy consumption ascribes to the communication module in cases (a), (b) and (d) of Fig. 5, where  $\beta_s \leq \beta_c$ . As we have concluded from Section 5.2, a variable communication radius does not cause any total power fluctuation as the overlap coefficient  $\alpha$  varies, which results the parallel curves in Fig. 5(a), (b) and (d). On the other hand, VRPM reduces the total energy dissipation on most occasions, even managing to boost performance by almost 20% as  $\beta$  increases.

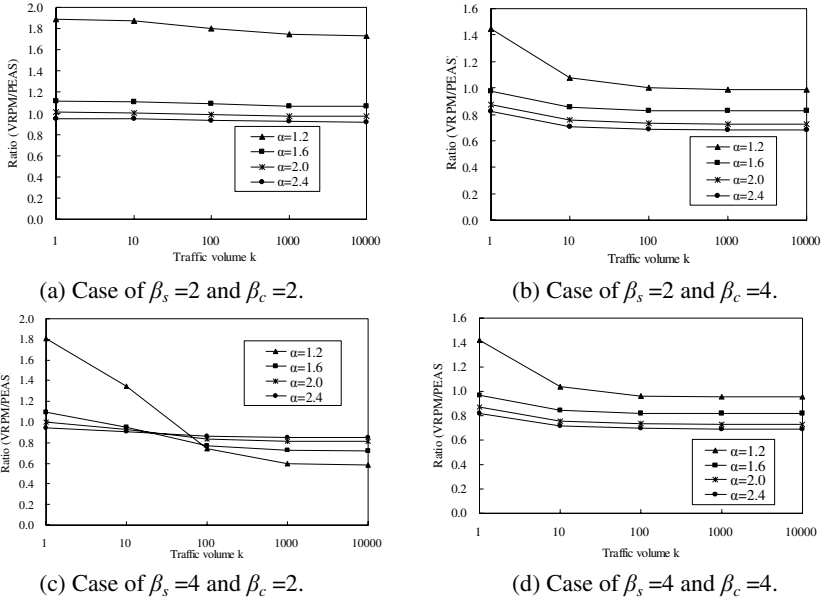


Fig. 5. Energy dissipation evaluations by the variation of parameter  $\beta$

In case (c), the sensing module comes to dominate the energy dissipation of the entire sensor network because  $\beta_s > \beta_c$  so that the overlap coefficient  $\alpha$  becomes the protagonist. Although VRPM with a small  $\alpha$  value cannot exceed PEAS at a low sensing traffic volume, it decreases the energy outstandingly as long as the sensing traffic becomes large enough. This implies that local sensing with a small sensing radius will be the optimal way in instances of high sensing traffic volumes.

### 5.3 Sensing/Communication Frequency

A simple sensing/communication frequency model mentioned at the beginning of Section 5 was applied in the above discussion. However, the sensing/communication

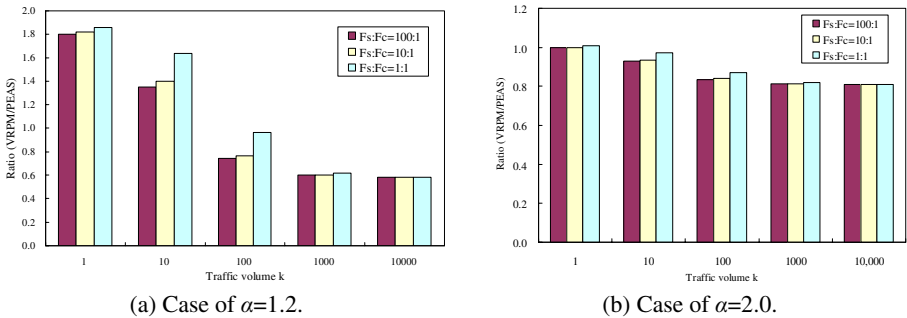


Fig. 6. Sensing/communication frequency evaluations



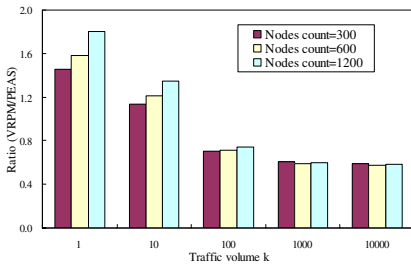
frequency will not be a constant in practice. Therefore, we define  $F_s$  and  $F_c$  as the sensing frequency (the number of sensing events per unit time) and the communication frequency (the number of communication events per unit time), respectively, so that  $F_s/F_c$  denotes the frequency proportion between sensing and communication. Energy dissipation is evaluated when  $F_s/F_c$  varies as 100:1, 10:1 and 1:1 (Fig. 6).

Evident fluctuations cannot be elicited from the above charts as  $F_s/F_c$  varies, which means that the time of sending sensed data does not affect the energy dissipation performance.

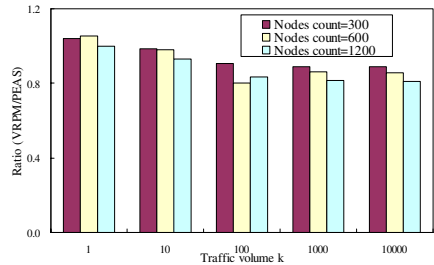
#### 5.4 Density of Sensor Nodes

Until now, all discussions have been based on a high-density sensor network. We deployed 1,200 sensor nodes to the  $100m \times 100m$  monitored region, giving a nodes density of 12 nodes/100m<sup>2</sup>. In this section, a low-density sensor network (3 nodes/100m<sup>2</sup>) and a medium-density sensor network (6 nodes/100m<sup>2</sup>) are evaluated to judge if VRPM is density independent.

Fig. 7 explains the comparison of three different density statuses. Whatever value the overlap coefficient  $\alpha$  has, VRPM provides a similar performance when the sensor node density varies from low to high, thus we can concluded that VRPM is not influenced by density.



(a) case of  $\alpha=1.2$ .



(b) Case of  $\alpha=2.0$ .

Fig. 7. Sensor node density evaluations

## 6 Conclusion and Future Work

The dominating energy consumption of active sensor networks lies in sensing modules and communication modules. VRPM is an efficient variable-radius energy conservation method that reduces energy redundancy in active sensor networks. The efficiency of energy conservation by VRPM was evaluated in several kinds of sensing and communication models to prove its broad applicability. Inside VRPM, an overlap coefficient  $\alpha$  is designed for sensor nodes to decide their appropriate radii according to their ambient situation. A self-adaptable algorithm based on our discussions on the overlap coefficient  $\alpha$  can be easily designed and embedded into every sensor node. We then proved that the energy consumption is independent of the sensing/communication frequency and density of sensor nodes through a series of simulations. Finally, we showed that VRPM can also maintain a geographical balance of

energy dissipation distribution, thereby extending the lifetime extension of the entire sensor network.

In future work, we plan to realize and evaluate VRPM on a real Mote<sup>®</sup> system.

## References

1. B. Yoshimi, and Y. Heights, "On Sensor Frameworks for Pervasive Systems", Workshop on Software Engineering for Wearable and Pervasive Computing (ICSE 2000), June 2000.
2. W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks", Proceedings of the 33rd International Conference on System Sciences (HICSS '00), January 2000.
3. K. Finkenzeller, "RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification", John Wiley & Sons; 2 Edition, May 9, 2003.
4. Z. Zhu, "RFID Analog Front-End Design Tutorial", Auto-ID lab at University of Adelaide.
5. F. Ye, G. Zhong, J. Cheng, S. Lu, and L. Zhang, "PEAS: A Robust Energy Conserving Protocol for Long-Lived Sensor Networks," Proc. IEEE Int'l Conf. Network Protocols (ICNP 2002), IEEE CS Press, pp. 200-201, 2002.
6. D. Tian and N.D. Georganas, "A Coverage-Preserving Node Scheduling Scheme for Large Wireless Sensor Networks", Proc. 1st ACM Workshop Wireless Sensor Networks and Applications, ACM Press, pp. 32-41, 2002.
7. V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava, "Energy-Aware Wireless Microsensor Networks", IEEE signal processing magazine, pp. 40-50, March 2002.
8. J. Wu and S. Yang, "Coverage Issue in Sensor Networks with Adjustable Ranges", ICPPW'04, August 15 - 18, 2004
9. Z. Zhou, S. Das, and Himanshu Gupta, "Variable-Radii Connected Sensor Cover in Sensor Networks", SECPM2004, October 4, 2004.
10. J. E. Wieselther, G. D. Nguyen, and A. Ephremides, "On the Construction of Energy-Efficient Broadcast and Multicast Trees in Wireless Networks", in Proceedings of the IEEE INFOCOM, 2000.
11. M. Cagalj, J. Hubaux, and C. Enz, "Minimum-energy Broadcast in All Wireless Networks: NP-completeness and Distributed Issues," in Proceedings of the International Conference on Mobile Computing and Networking (MobiCom), 2002.

# Automata-Theoretic Performance Analysis Method of Soft Real-Time Systems

Satoshi Yamane

Graduate School of Natural Science, Kanazawa University,  
Kanazawa City, Japan, Zip/Code 920-1192  
syamane@is.t.kanazawa-u.ac.jp

**Abstract.** Recently, the verification method of schedulability of real-time operating systems using timed automata have been developed. On the other hand, as soft real-time systems such as distributed systems and multimedia systems have been increasing, it is important to design soft real-time systems. Especially, performance analysis methods are important for soft real-time systems. In this paper, we develop the automata-theoretic performance analysis method of soft real-time systems by extending the verification method of schedulability of hard real-time systems using utility functions.

## 1 Introduction

Real-time systems are of vital economic importance and are literally becoming ubiquitous. They have already become an integral component of safety critical systems involving aviation, telecommunications, and process control applications. It is important to formally specify and verify real-time systems. Real-time systems are characterized as those where the correctness of applications depends not only on the correctness of the logical computation being performed, but also on the time at which the results are produced [1]. Real-time systems are classified into hard real-time systems and soft real-time systems [2]. Hard real-time systems do not tolerate any missed deadlines. This requires that the task scheduling is based on worst-case task execution time estimates. This leads to inefficient utilization of resources, particularly when tasks normally require much less time to complete execution than the worst case estimates, and the worst case behavior is very rare. On the other hand, in soft real-time systems, where some missed deadlines can be tolerated, it is not necessary to use worst case execution time estimates. The probability that all tasks in the system complete by a specified deadline can be increased by allowing a small amount of slack times. The slack time is to tolerate the failure of some tasks completing within their assigned execution times. Thus trade-off between the guarantee that all tasks meet their deadlines and the cost of such guarantees can be made.

In hard real-time systems, it is important to verify whether worst-case task execution time is less than deadline or not. This verification is called schedulability [1]. On the other hand, in soft real-time systems, the performance of a

scheduling algorithm can be measured by accumulating the values of the task utility functions computed at their complete time [3]. Today, timed automaton [4] is the standard tool for specifying and verifying real-time systems by model-checking methods [5, 6]. Moreover, recently, the schedulability checking problem can be solved using extended timed automata by Wang Yi [7, 8]. Before Wang Yi's study [8], the preemptive schedulability checking problem has been able to be solved using only stopwatch automata [9, 10, 11, 12, 13].

On the other hand, in soft real-time systems, the performance of a scheduling algorithm has been measured by accumulating the values of the task utility functions computed at their complete time or determining lower bounds on the frequency of missed deadlines as follows. (1) In 1985, E.D. Jensen has used value functions to describe the performance of existing process scheduling algorithms [14]. (2) In 1995, B.Kao has studied the performance of distributed soft real-time systems that use standard components with various scheduling algorithms and have suggested ways to improve them [15]. (3) In 1994, K.M. Kavi has studied a simple model that combines the failure to meet deadlines with hardware/software failures in computing the reachability of a real-time systems. They have defined the performability as the probability of meeting deadlines by real-time tasks in the presence of hardware and software failures. (4) In 1999, M.K. Gardner and J.W.S. Liu have studied an analysis technique by which a lower bound on the percentage of deadlines that a periodic task meets is determined and compare the lower bound with simulation results for an example system [17].

On the other hand, some researchers have extended existing languages with probabilities such as probabilistic process algebra [18], probabilistic timed automaton with continuous probabilities [19, 21], probabilistic timed automaton with discrete probabilities [20]. But they have not studied performance analysis.

In this paper, we propose an automata-theoretic performance analysis method that combines timed automata with utility functions. More practically speaking, we extend Wang Yi's timed automata [8] with utility functions. While existing studies have been based on simulation methods, our proposed method is based on mathematical model. Using our proposed method, we can formally evaluate schedulers of soft real-time systems such as multimedia systems and distributed systems. Our proposed method is different from Wang Yi's method [8] in the following two important points:

1. In Wang Yi's method, task is characterized as the execution time and the deadline. On the other hand, in our proposed method, task is characterized as the execution time, the deadline and the utility function.
2. In Wang Yi's method, the schedulability problem can be transformed to a reachability problem for timed automata. On the other hand, in our proposed method, the performance analysis can be transformed to a brute force search for timed computation tree of timed automata and a calculation of the value of every state.

The paper is organized as follows: In section 2, we propose extended timed automata for performance analysis of soft real-time systems. In section 3, we propose our automata-theoretic performance analysis method. In section 4, we

propose algorithms of our automata-theoretic performance analysis. Finally, in section 5, we present conclusions.

## 2 Timed Automata of Performance Analysis

In this paper, we propose extended timed automata for performance analysis of soft real-time systems.

### 2.1 Soft Real-Time Systems

First we define soft real-time systems. In soft real-time systems, where some missed deadlines can be tolerated, it is not necessary to use worst case execution time estimates. The probability that all tasks in the system complete by a specified deadline can be increased by allowing a small amount of slack times. According to the probability, the value of a task decreases. The task importance can be better described by an utility function. Figure 1 illustrates some utility functions that can be associated with tasks in order to describe their importance. A hard task contributes to a value only if it completes within its deadline  $D$ , and since a deadline miss would jeopardize the behavior of the whole system, the value  $v(t)$  after its deadline  $D$  can be considered minus infinity in many situations. A task with a soft deadline, instead, can still give a value  $v(t)$  to the system if executed after its deadline  $D$ , although this value may decrease with time.

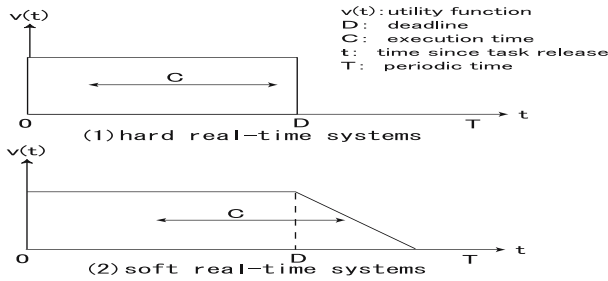


Fig. 1. Utility function that can be associated to a task to describe its importance

### 2.2 Syntax of Extended Timed Automata

First, we define task as follows.

#### Definition 1 (Task)

Let  $\mathcal{P}$  ranged over by  $P$  and  $Q, R$  etc, denote a finite set of task types. A task type may have different instances that are copies of the same program with different inputs. Each task  $P$  is characterized as a pair of natural numbers denoted  $P(D_P, C_P, v_P(t_P))$  with  $C_P \leq D_P$ , where  $C_P$  is the execution time (or computation time) of  $P$ ,  $D_P$  is the deadline for  $P$ ,  $v_P(t_P)$  is the utility function. ■

Next, we introduce a preliminary.

**Definition 2 (Preliminary)**

We introduce a preliminary.

1. *Act* is a finite alphabet for actions. Let  $Act = \{a, b, c, \dots\}$ .
2.  $\mathcal{C}$  is a finite set of real-valued variables. Let  $\mathcal{C} = \{x_1, x_2, x_3, \dots\}$ .
3. We use  $\mathcal{B}(\mathcal{C})$  ranged over by  $g$  to denote the set of conjunctive formulas of atomic constraints in the form:  $x_i \sim d_i$  or  $x_i - x_j \sim d_{ij}$ , where  $x_i, x_j \in \mathcal{C}$ ,  $\sim \in \{\leq, <, \geq, >\}$ ,  $d_i, d_{ij}$  are natural numbers. ■

Next, we define extended timed automaton.

**Definition 3 (Syntax of extended timed automaton)**

An extended timed automaton  $\mathbf{A}$  is a tuple  $(N, l_0, E, I, M)$ , where

1.  $N$  is a finite set of locations ranged over by  $l, m, n$
2.  $l_0 \in N$  is the initial location
3.  $E \subseteq N \times \mathcal{B}(\mathcal{C}) \times Act \times 2^{\mathcal{C}} \times N$  is the set of edges
4.  $I : N \hookrightarrow \mathcal{B}(\mathcal{C})$  is a function assigning each location with a clock constraint
5.  $M : N \hookrightarrow \mathcal{P}$  is a partial function assigning locations with tasks ■

Next we show some simple example of a task model.

**Example 1 (A simple example of a task model)**

Tasks are classified into periodic tasks, aperiodic tasks, sporadic tasks [22]. In the periodic task model, each computation or data transmission that is executed repeatedly at regular or semiregular time intervals in order to provide a function of the system on a continuing basis is modeled as a periodic task. On the other hand, a task is aperiodic if the jobs in the task have either soft deadlines or no deadlines. Moreover, a task is sporadic if the jobs in the task are sporadically released. We show examples of periodic tasks, aperiodic tasks, sporadic tasks, and specify them using extended timed automata in Figure 2. ■

**2.3 Semantics of Extended Timed Automata**

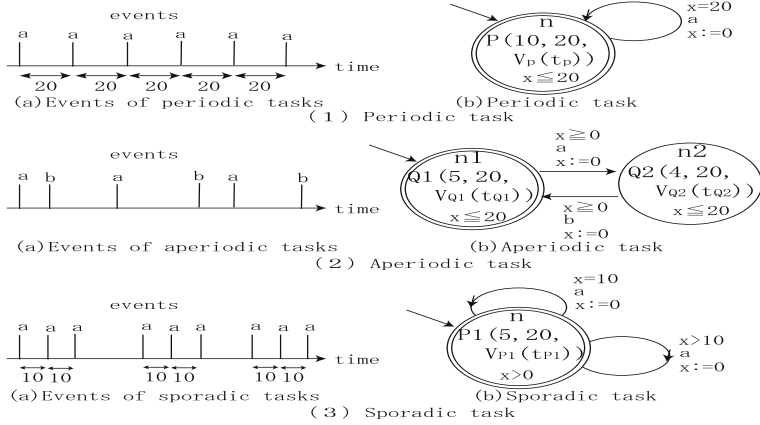
Next, we define semantics of extended timed automata.

Extended timed automata perform two types of transitions. Delay transitions correspond to the execution of running tasks with high priority and idling for the other tasks waiting to run. Discrete transitions correspond to the arrival of new task instances.

First, we define semantic states as follows:

**Definition 4 (Semantic states of timed automaton)**

We represent the values of clocks as functions from  $C$  to the non-negative reals  $\mathbf{R}$ .  $\mathcal{V}$  denotes the set of assignments of  $C$ . A semantic state of an extended timed automaton is a triple  $(l, u, q)$ , where  $l \in N$  is the current node,  $u \in \mathcal{V}$  denotes the current values of clocks, and  $q$  is the current task queue. We assume that the task queue takes the form:



**Fig. 2.** Examples of periodic tasks, aperiodic tasks, sporadic tasks

$$[P_1(c_{P_1}, d_{P_1}, v_{P_1}(t_{P_1})), \dots, P_n(c_{P_n}, d_{P_n}, v_{P_n}(t_{P_n}))],$$

where  $P_i(c_{P_i}, d_{P_i}, v_{P_i}(t_{P_i}))$  denotes a released instance of task type  $P_i$  with remaining computing time  $c_{P_i}$ , relative deadline  $d_{P_i}$  and value  $v_{P_i}(t_{P_i})$ . Here  $t_{P_i}$  denotes the elapsed time since the release of  $P_i$ . ■

A scheduling strategy **Sch** such as FPS (Fixed Priority Scheduling) and EDF (Earliest Deadline First) is a sorting function which changes the ordering of the task queue elements according to the task parameters. A delay transition with  $t$  time units is to execute the task in the first position of the queue with  $t$  time units. Thus the delay transition will decrease the computing time of the first task by  $t$  and increase the elapsed time  $t_P$  by  $t$ , and compute  $v_P(t_P)$ . If the computation time becomes 0, the task should be removed from the queue. We define the followings:

1. **Sch** is a sorting function which changes the ordering of the task queue elements.  
For example,  $\text{EDF}([P(3.1, 10, v_p(t_p)), Q(4, 5.3, v_q(t_q))]) = [Q(4, 5.3, v_q(t_q)), P(3.1, 10, v_p(t_p))]$ .
2. **Run** is a function which given a real number  $t$  and a task  $q$  returns the resulted task queue after  $t$  time units of execution according to available computing resources.  
For example, let  $q = [Q(4, 5, v_q(t_q)), P(3, 10, v_p(t_p))]$ . Then  $\text{Run}(q, 6) = [P(1, 4, v_p(t_p + 6))]$  and, the value of  $Q$  is  $v_q(t_q + 4)$ .

Next we define  $u \models g, u + t, u[r \mapsto 0]$ ,  $x := x - c$  in order to define the operational semantics of timed automaton.

1. We use  $u \models g$  to denote that the clock assignment  $u$  satisfies the constraint  $g$ .
2. We use  $u + t$  to denote the clock assignment which maps each  $x$  to the value  $u(x) + t$ .

3. We use  $u[r \mapsto 0]$  for  $r \subseteq \mathcal{C}$  to denote the clock assignment which each clock in  $r$  to 0 and agrees with  $u$  for the other clocks.
4. We use  $x := x - c$  to denote the subtraction of  $c$  from  $x$ .

We define the operational semantics of extended time automata as follows:

**Definition 5 (Operational semantics)**

Given a scheduling strategy **Sch**, the semantics of an extended timed automaton  $\mathbf{A} = (N, l_0, E, I, M)$  with an initial state  $(l_0, u_0, q_0)$  is a transition system defined by the following rules:

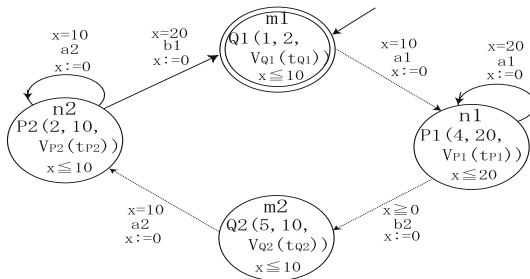
1. discrete transition:  
 $(l, u, q) \xrightarrow{g, a, r}_{Sch} (m, u[r \mapsto 0], \mathbf{Sch}(M(m) :: q))$  if  $l \xrightarrow{g, a, r} m$  and  $u \models g$
2. delay transition:  
 $(l, u, q) \xrightarrow{t}_{Sch} (l, u + t, \mathbf{Run}(q, t))$  if  $(u + t) \models I(l)$ ,  
 where  $t$  is assigned to the maximum delay as we execute maximum numbers of tasks.

Here  $M(m) :: q$  denotes the queue  $q$  with  $M(m)$  inserted into it. ■

We show an example as follows:

**Example 2 (Example of behavior of tasks)**

First we show an example of a task model.  $P_1$  and  $P_2$  are periodic tasks, and  $Q_1$  and  $Q_2$  are triggered by events. First, a task  $Q_1$  is triggered in node  $m_1$ . The automaton models a system starting in node  $m_1$ , which moves to node  $n_1$  by event  $a_1$  at 10 time, which triggers the task  $P_1$ . Moreover, the system periodically triggers the task  $P_1$  at 20 time in node  $n_1$ . After that, the system moves to node  $m_2$  by event  $b_2$  at any time, which triggers the  $Q_2$ .



**Fig. 3.** Example of an extended timed automaton

Assume that preemptive earliest deadline(EDF) is used to schedule the task queue. Then the automaton with initial state may demonstrate the following sequence of typical transitions:



$$\begin{aligned}
 & (m_1, [x = 0], [Q_1(1, 2, v_{Q_1}(0)]) \\
 \xrightarrow{1}_{Sch} & (m_1, [x = 1], [Q_1(0, 1, v_{Q_1}(1))]) = (m_1, [x = 1], []) \\
 \xrightarrow{9}_{Sch} & (m_1, [x = 10], []) \\
 \xrightarrow{a_1}_{Sch} & (n_1, [x = 0], [P_1(4, 20, v_{P_1}(0)]) \\
 \xrightarrow{4}_{Sch} & (n_1, [x = 4], [P_1(0, 16, v_{P_1}(4))]) = (n_1, [x = 4], []) \\
 & \dots\dots\dots \\
 & \dots\dots\dots
 \end{aligned}$$



### 3 Automata-Theoretic Performance Analysis Method

In this section, we propose an automata-theoretic performance analysis method of soft real-time systems. In 2002, Wang Yi and his colleagues have shown that schedulability checking problem based on timed automata is decidable [7, 8]. In this paper, we propose an automata-theoretic performance analysis method of soft real-time systems by extending Wang Yi’s method. In Wang Yi’s method, the schedulability problem can be transformed to a reachability problem for timed automata. On the other hand, in our proposed method, the performance analysis can be transformed to a brute force search for timed computation tree of timed automata and a calculation of the value of every state. In general, as real-time systems is nondeterministic, we must search for timed computation tree [5].

We realize an automata-theoretic performance analysis method by the followings:

For a given timed automaton, we construct a timed computation tree with root state  $(l_0, u_0, q_0)$  and other states such as  $(l_k, u_k, q_k)(k \geq 1)$ . At the same time, we compute the value of  $(l_k, u_k, q_k)$ . Finally, we sum up the values per each deterministic behavior.

First, we define the notion of timed sequence of extended timed automaton.

**Definition 6 (Timed sequence)**

We define the notion of timed sequence of extended timed automaton as follows:

For an automaton with initial state  $(l_0, u_0, q_0)$ , timed sequence is as follows:

$$\omega = (l_0, u_0, q_0) \xrightarrow{t_0} \dots \xrightarrow{g_{i-1}, a_{i-1}, r_{i-1}} (l, u, q) \xrightarrow{t_i} \dots$$

In general, one extended timed automaton has a number of timed sequences.



Next, we define the notion of reachability as for extended timed automaton.

**Definition 7 (Reachability)**

We define the notion of reachability as for extended timed automaton as follows:

For an automaton with initial state  $(l_0, u_0, q_0)$ ,  $(l, u, q)$  is reachable

$$\omega = (l_0, u_0, q_0) \xrightarrow{t_0} \dots \xrightarrow{g_{i-1}, a_{i-1}, r_{i-1}} (l, u, q)$$

or

$$\omega = (l_0, u_0, q_0) \xrightarrow{t_0} \dots \xrightarrow{t_{i-1}} (l, u, q).$$



Next we define an automata-theoretic performance analysis method of soft real-time systems.

**Definition 8 (An Automata-Theoretic Performance Analysis Method)**

For a given timed automaton, we construct a timed computation tree with root state  $(l_0, u_0, q_0)$  and other states such as  $(l_k, u_k, q_k) (k \geq 1)$ . At the same time, we compute the value of  $(l_k, u_k, q_k) (k = 0, 1, 2, \dots)$  as follows:

where

$$q_k = [P_1^k(c_{P_1}^k, d_{P_1}^k, v_{P_1}(t_{P_1}^k)), \dots, P_n^k(c_{P_n}^k, d_{P_n}^k, v_{P_n}(t_{P_n}^k))].$$

1. For each deterministic behavior, from each initial state, we construct the timed sequence consisting of the set of semantic states, and compute values as follows:

- (a) Case of  $t_{P_i}^k > T_{P_i}$ :

System error occurs,

where  $t_{P_i}^k$  is the elapsed time since release of  $P_i$ , and  $T_{P_i}$  is the period or the constraint of  $P_i$ .

- (b) Case of  $t_{P_i}^k \leq T_{P_i}$ :

We compute the sum of values as follows:

For  $\forall i$ , we compute the value using  $v_{P_i}(t_{P_i}^k)$  at  $c_i^k = 0$ :

$$\sum_{i=1}^n v_{P_i}(t_{P_i}^k)$$

2. We repeat the above procedure 1., and compute the set of sums. For example, the sum is  $\sum_{i=1}^n v_{P_i}(t_{P_i}^k)$ . Finally, the number of sums, which we have computed, is equal to the number of nondeterministic behaviors. Each deterministic behavior has each sum. ■

Finally, we mention that an automata-theoretic performance analysis method is decidable.

**Theorem 1 (Decidability of an automata-theoretic performance analysis)**

The problem of computing an automata-theoretic performance analysis for extended timed automaton is decidable.

**Proof 1.** Our extended timed automaton is the class of bounded timed automaton with subtraction [8]. Thus the problem of computing performance analysis for extended timed automaton is decidable. ■

## 4 Implementation

In this section, we show the implementation of an automata-theoretic performance analysis method.

We compute the performance in **Definition 8** according to semantic transitions in **Definition 5**. In this performance analysis method, it is important

to compare elapsed time with remaining computing time, and compute values if remaining computing time becomes zero. We use DBMs(Difference Bounds Matrices) [23, 24] in order to compute remaining computing time.

First, we define DBMs as follows.

**Definition 9 (DBMs)**

*Suppose the extended timed automaton has  $n$  clocks,  $x_1, \dots, x_n$ . Then a clock zone is represented by a  $(n + 1) \times (n + 1)$  matrix  $D$ . For each  $i$ , the entry  $D_{i0}$  gives an upper bound on the clock  $x_i$ , and the entry  $D_{0i}$  gives a lower bound on the clock  $x_i$ . For every pair  $i, j$ , the entry  $D_{ij}$  gives an upper bound on the difference of the clocks  $x_i$  and  $x_j$ . To distinguish between a strict and a nonstrict bound (i.e. to distinguish between constraints such as  $x < 2$  and  $x \leq 2$ ), and allow for the possibility of absence of a bound, define the bounds-domain to be  $\mathbf{Z} \times \{<, \leq\} \cup \{\infty\}$ . The constant  $\infty$  denotes the absence of a bound, the bound  $(c, \leq)$  denotes  $\leq c$ , the bound  $(c, <)$  denotes  $< c$ . ■*

Next we define Operations of DBMs.

**Definition 10 (Operations of DBMs)**

*In this paper, we use the following operations:*

1. *Intersection:*

*The intersection of two DBMs  $D$  and  $D'$  can easily be computed from two DBMs. The intersection DBMs  $D \cap D' = D \cap D'$  is the canonical form of DBMs  $D \cap D'$ , which is computed by the followings:*

$$D_{ij} \cap D'_{ij} = \min\{D_{ij}, D'_{ij}\}$$

2. *Time successor:*

*$D \nearrow$  is obtained from  $D$  by removing all inequalities that place upper bounds on the absolute values of the clocks, i.e. inequalities of the form  $x_i - x_0 \leq d_{i0}$  or  $x_i - x_0 < d_{i0}$ .*

3. *Reset:*

*$D[r := 0]$  is obtained from  $D$  by first ignoring all constraints on variables in  $r$ , and then taking the subset for which all variables in  $r$  equal to zero. ■*

Next we define the method of computing task queue.

**Definition 11 (Computing task queue by DBMs)**

*For state  $(l_k, u_k, q_k)$  ( $k = 0, 1, 2, \dots$ ), we define the method of computing such task queue as*

$$q_k = [P_1^k(c_{P_1^k}, d_{P_1^k}, v_{P_1}(t_{P_1^k})), \dots, P_n^k(c_{P_n^k}, d_{P_n^k}, v_{P_n}(t_{P_n^k}))].$$

*We compute the task queue by dividing transitions into delay and discrete transition.*

**Delay transition:**

*First we compute the elapsed time in the state  $(l_k, u_k, q_k)$ . Assume that DBMs is  $D$  in the entrance of the state, where all the clocks are zero in an initial state. Here we introduce the elapsed timer  $t_{l_k}$ .  $D$  becomes  $D'$  by the fact that  $t_{l_k}$  is*

added to  $D$ . As the minimum clock is assigned to  $t_{l_k}$ , the following entries are added to  $D$ .

$$t_{l_k} - x_0 \leq 0, t_{l_k} - x_1 \leq 0, \dots, t_{l_k} - x_n \leq 0, x_0 - t_{l_k} \leq 0$$

We compute the elapsed time in state  $(l_k, u_k, q_k)$  based on  $DI$ .

$$(((DI \cap D_{I(l_k)}) \wedge) \cap D_{I(l_k)}) \cap D_g,$$

where  $D_{I(l_k)}$  is DBMs obtained from clock constraint in node  $l_k$ ,  $D_g$  is DBMs obtained from  $g$ .

Now we compute  $(l_k, u_k + t_{l_k}, \mathbf{Run}(q_k, t_{l_k}))$  using  $t_{l_k}$ .

In the followings, we compute  $\mathbf{Run}(q_k, t_{l_k})$  by dividing  $t_{l_k}$  into various cases.  $t_{l_k}$  is divided into  $t_{l_k} \leq d_{l_k}$ ,  $t_{l_k} = d_{l_k}$ ,  $t_{l_k} < d_{l_k}$ ,  $t_{l_k} > d_{l_k}$ ,  $t_{l_k} \geq d_{l_k}$ . Moreover, according to relations between  $d_{l_k}$  and  $c_{P_1}^k$ ,  $c_{P_1}^k$  is divided into  $c_{P_1}^k \leq d_{l_k}$ ,  $c_{P_1}^k = d_{l_k}$ ,  $c_{P_1}^k < d_{l_k}$ ,  $c_{P_1}^k > d_{l_k}$ ,  $c_{P_1}^k \geq d_{l_k}$ .

We show these cases by the Figure 4.

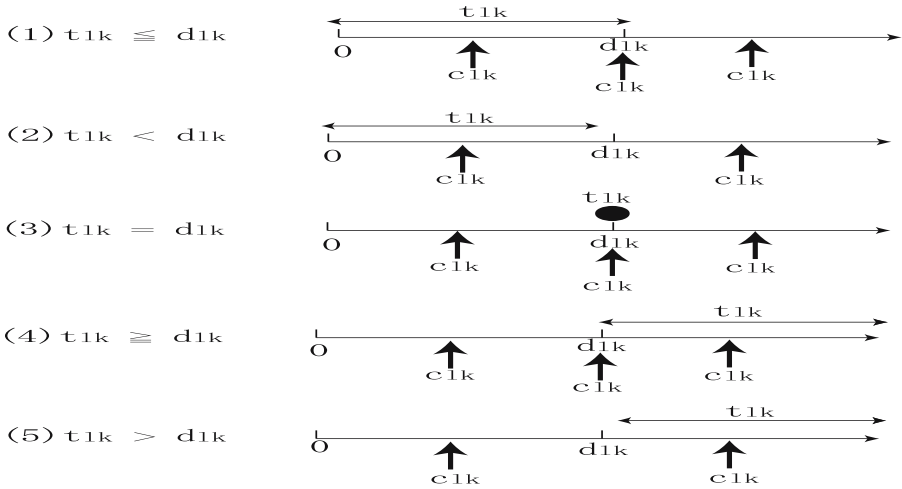


Fig. 4. Cases of  $t_{l_k}$

In the followings, we compute  $\mathbf{Run}(q_k, t_{l_k})$  according to the above cases. Here task  $P_i$  has an error, if  $t_{l_k}$  is greater than  $T_{P_i}$ .

1.  $t_{l_k} \leq d_{l_k}$ :

Then we consider three cases such as  $c_{P_i}^k < d_{l_k}$ ,  $c_{P_i}^k = d_{l_k}$ ,  $c_{P_i}^k > d_{l_k}$ .

(a)  $c_{P_1}^k < d_{l_k}$

In this case, task queue is as follows:

$$\mathbf{Run}(q_k, t_{l_k}) = [ P_1^k(0, d_{P_1}^k - c_{P_1}^k, v_{P_1}(t_{P_1}^k + c_{P_1}^k)), \dots, P_i^k(0, d_{P_i}^k - (c_{P_1}^k + \dots + c_{P_{i-1}}^k), v_{P_i}(t_{P_i}^k + (c_{P_1}^k + \dots + c_{P_{i-1}}^k))),$$

$$P_{i+1}^k(c_{P_{i+1}}^k - (d_{l_k} - (c_{P_1}^k + \dots + c_{P_i}^k)), d_{P_i}^k - d_{l_k}, v_{P_i}(t_{P_{i+1}}^k + d_{l_k})), \\ \dots, \\ P_n^k(c_{P_n}^k, d_{P_n}^k - d_{l_k}, v_{P_n}(t_{P_n}^k + d_{l_k}))), \text{ where } t_{l_k} = d_{l_k}.$$

Therefore we compute the value as follows:

$$v_{P_1}(t_{P_1}^k + c_{P_1}^k)) + v_{P_2}(t_{P_2}^k + c_{P_1}^k + c_{P_2}^k)) + \dots$$

(b)  $c_{P_1}^k = d_{l_k}$ :

In this case, task queue is as follows:

$$\mathbf{Run}(q_k, t_{l_k}) = [ \\ P_1^k(0, d_{P_1}^k - c_{P_1}^k, v_{P_1}(t_{P_1}^k + c_{P_1}^k)), \\ P_2^k(c_{P_2}^k, d_{P_2}^k - c_{P_1}^k, v_{P_2}(t_{P_2}^k + d_{l_k})), \\ \dots, \\ P_n^k(c_{P_n}^k, d_{P_n}^k, v_{P_n}(t_{P_n}^k + d_{l_k}))), \text{ where } t_{l_k} = d_{l_k}.$$

Therefore we compute the value as follows:

$$v_{P_1}(t_{P_1}^k + c_{P_1}^k))$$

(c)  $c_{P_1}^k > d_{l_k}$ :

In this case, task queue is as follows:

$$\text{As } \mathbf{Run}(q_k, t_{l_k}) = [ \\ P_1^k(c_{P_1}^k - d_{l_k}, d_{P_1}^k - d_{l_k}, v_{P_1}(t_{P_1}^k + d_{l_k})), \\ P_2^k(c_{P_2}^k, d_{P_2}^k, v_{P_2}(t_{P_2}^k + d_{l_k})), \\ \dots, \\ P_n^k(c_{P_n}^k, d_{P_n}^k, v_{P_n}(t_{P_n}^k + d_{l_k}))).$$

we can not compute the value, where  $t_{l_k} = d_{l_k}$ .

2.  $t_{l_k} < d_{l_k}$ :

In this case, we consider two cases such as  $c_{P_i}^k < d_{l_k}$ ,  $c_{P_i}^k \geq d_{l_k}$ .

(a)  $c_{P_1}^k < d_{l_k}$ :

In this case, task queue is as follows:

$$\mathbf{Run}(q_k, t_{l_k}) = [ \\ P_1^k(0, d_{P_1}^k - c_{P_1}^k, v_{P_1}(t_{P_1}^k + c_{P_1}^k)), \\ \dots, \\ P_i^k(0, d_{P_i}^k - (c_{P_1}^k + \dots + c_{P_{i-1}}^k), v_{P_i}(t_{P_i}^k + (c_{P_1}^k + \dots + c_{P_{i-1}}^k))), \\ P_{i+1}^k(c_{P_{i+1}}^k - (d_{l_k}^- - (c_{P_1}^k + \dots + c_{P_i}^k)), d_{P_i}^k - d_{l_k}^-, v_{P_i}(t_{P_{i+1}}^k + \\ d_{l_k}^-)), \\ \dots, \\ P_n^k(c_{P_n}^k, d_{P_n}^k - d_{l_k}^-, v_{P_n}(t_{P_n}^k + d_{l_k}^-))), \text{ where } d_{l_k}^- \text{ is almost equal}$$

to  $d_{l_k}$  and less than  $d_{l_k}$ .

Here  $t_{l_k} = d_{l_k}^-$ .

Therefore we compute the value as follows:

$$v_{P_1}(t_{P_1}^k + c_{P_1}^k)) + v_{P_2}(t_{P_2}^k + c_{P_1}^k + c_{P_2}^k)) + \dots$$

(b)  $c_{P_1}^k \geq d_{l_k}$

In this case, task queue is as follows:

$$\text{As } \mathbf{Run}(q_k, t_{l_k}) = [ \\ P_1^k(c_{P_1}^k - d_{l_k}^-, d_{P_1}^k - d_{l_k}^-, v_{P_1}(t_{P_1}^k + d_{l_k}^-)), \\ P_2^k(c_{P_2}^k, d_{P_2}^k - d_{l_k}^-, v_{P_2}(t_{P_2}^k + d_{l_k}^-)), \\ \dots, \\ P_n^k(c_{P_n}^k, d_{P_n}^k - d_{l_k}^-, v_{P_n}(t_{P_n}^k + d_{l_k}^-))), \text{ we can not compute the}$$

value.

Here  $t_{l_k} = d_{l_k}^-$ .

3.  $t_{l_k} = d_{l_k}$ :

In this case, we consider three cases such as  $c_{P_i}^k < d_{l_k}$ ,  $c_{P_i}^k = d_{l_k}$ ,  $c_{P_i}^k > d_{l_k}$ .

(a)  $c_{P_1}^k < d_{l_k}$ :

In this case, task queue is as follows:

$$\text{Run}(q_k, t_{l_k}) = [ P_1^k(0, d_{P_1}^k - c_{P_1}^k, v_{P_1}(t_{P_1}^k + c_{P_1}^k)), \dots, P_i^k(0, d_{P_i}^k - (c_{P_1}^k + \dots + c_{P_{i-1}}^k), v_{P_i}(t_{P_i}^k + (c_{P_1}^k + \dots + c_{P_{i-1}}^k))), P_{i+1}^k(c_{P_{i+1}}^k - (d_{l_k} - (c_{P_1}^k + \dots + c_{P_i}^k)), d_{P_i}^k - d_{l_k}, v_{P_i}(t_{P_{i+1}}^k + d_{l_k})), \dots, P_n^k(c_{P_n}^k, d_{P_n}^k - d_{l_k}, v_{P_n}(t_{P_n}^k + d_{l_k}))].$$

Here  $t_{l_k} = d_{l_k}$ .

Therefore we compute the value as follows:

$$v_{P_1}(t_{P_1}^k + c_{P_1}^k) + v_{P_2}(t_{P_2}^k + c_{P_1}^k + c_{P_2}^k) + \dots$$

(b)  $c_{P_1}^k = d_{l_k}$

In this case, task queue is as follows:

$$\text{Run}(q_k, t_{l_k}) = [ P_1^k(0, d_{P_1}^k - c_{P_1}^k, v_{P_1}(t_{P_1}^k + c_{P_1}^k)), P_2^k(c_{P_2}^k, d_{P_2}^k - c_{P_1}^k, v_{P_2}(t_{P_2}^k + d_{l_k})), \dots, P_n^k(c_{P_n}^k, d_{P_n}^k, v_{P_n}(t_{P_n}^k + d_{l_k}))].$$

Here  $t_{l_k} = d_{l_k}$ .

Therefore we compute the value as follows:

$$v_{P_1}(t_{P_1}^k + c_{P_1}^k)$$

(c)  $c_{P_1}^k > d_{l_k}$ :

In this case, task queue is as follows:

$$\text{As Run}(q_k, t_{l_k}) = [ P_1^k(c_{P_1}^k - d_{l_k}, d_{P_1}^k - d_{l_k}, v_{P_1}(t_{P_1}^k + d_{l_k})), P_2^k(c_{P_2}^k, d_{P_2}^k - d_{l_k}, v_{P_2}(t_{P_2}^k + d_{l_k})), \dots, P_n^k(c_{P_n}^k, d_{P_n}^k - d_{l_k}, v_{P_n}(t_{P_n}^k + d_{l_k}))],$$

we can not compute the value.

Here  $t_{l_k} = d_{l_k}$ .

4.  $t_{l_k} \geq d_{l_k}$ :

In this case, although  $c_{P_i}^k$  is less or greater than  $d_{l_k}$ , task queue is as follows:

$$\text{Run}(q_k, t_{l_k}) = [ P_1^k(0, d_{P_1}^k - c_{P_1}^k, v_{P_1}(t_{P_1}^k + c_{P_1}^k)), P_2^k(0, d_{P_2}^k - (c_{P_1}^k + c_{P_2}^k), v_{P_2}(t_{P_2}^k + c_{P_1}^k + c_{P_2}^k)), \dots, P_n^k(0, d_{P_n}^k - (c_{P_1}^k + \dots + c_{P_n}^k), v_{P_n}(t_{P_n}^k + c_{P_1}^k + \dots + c_{P_n}^k))].$$

Here  $t_{l_k} = c_{P_1}^k + \dots + c_{P_n}^k$ .

Therefore we compute the value as follows:

$$v_{P_1}(t_{P_1}^k + c_{P_1}^k) + \dots + v_{P_n}(t_{P_n}^k + c_{P_1}^k + \dots + c_{P_n}^k)$$

5.  $t_{l_k} > d_{l_k}$ :

In this case, although  $c_{P_i}^k$  is less or greater than  $d_{l_k}$ , task queue is as follows:

$$\mathbf{Run}(q_k, t_{l_k}) = [$$

$$P_1^k(0, d_{P_1}^k - c_{P_1}^k, v_{P_1}(t_{P_1}^k + c_{P_1}^k)),$$

$$P_2^k(0, d_{P_2}^k - (c_{P_1}^k + c_{P_2}^k), v_{P_2}(t_{P_2}^k + c_{P_1}^k + c_{P_2}^k)),$$

$$\dots, \dots,$$

$$P_n^k(0, d_{P_n}^k - (c_{P_1}^k + \dots + c_{P_n}^k), v_{P_n}(t_{P_n}^k + c_{P_1}^k + \dots + c_{P_n}^k))].$$

Here  $t_{l_k} = c_{P_1}^k + \dots + c_{P_n}^k$ .

Therefore we compute the value as follows:

$$v_{P_1}(t_{P_1}^k + c_{P_1}^k) + \dots + v_{P_2}(t_{P_2}^k + c_{P_1}^k + \dots + c_{P_n}^k)$$

**Discrete transition:**

After delay transition, as  $l \xrightarrow{g,a,r} m$  and  $u \models g$  hold true, the following transition occurs:

$$(l, u, q) \xrightarrow{g,a,r}_{Sch} (m, u[r \mapsto 0], \mathbf{Sch}(M(m) :: q)),$$

where  $\mathbf{Sch}(M(m) :: q)$  can be obtained by adding the task in node  $m$ .

For each deterministic behavior, by repeating 1. and 2. from an initial node, we can compute  $\sum_{i=1}^n v_{P_i}(t_{P_i}^k)$ .

Therefore, we can get the sum of values for each deterministic behavior. ■

## 5 Conclusion

In this paper, we develop an automata-theoretic performance analysis method of soft real-time systems by extending the verification method of schedulability of hard real-time systems using utility functions. In our proposed method, task is characterized as the execution time, the deadline and the utility function. Moreover, performance analysis can be transformed to a brute force search for timed computation tree of timed automata and a calculation of the value of every state. We have implemented an automata-theoretic performance analyzer on Sun Blade2000. Now, we are going to apply our proposed method into practical problems.

## References

1. M. Joseph, editor. Real-time Systems: Specification, Verification and Analysis. *Prentics-Hall*, 1996.
2. Jane W. S. Liu. Real-Time Systems. *Prentics-Hall*,2000.
3. Giorgio C. Buttazzo. Hard Real-Time Computing Systems : Predictable Scheduling Algorithms and Applications. *Kluwer*, 1997.
4. R. Alur, D.L. Dill. A theory of timed automata. *TCS*,Vol. 126, pp.183-235, 1994.
5. R. Alur, C. Courcoubetis, D.L. Dill. Model-Checking in Dense Real-Time. *Information and Computation*, Vol. 104, pp. 2-34, 1993.
6. T. A. Henzinger, X. Nicollin, J. Sifakis, S. Yovine. Symbolic model checking for real-time systems. *Information and Computation*, Vol. 111, pp. 193-244, 1994.

7. C. Ericsson, A. Wall, Wang Yi. Timed Automata as Task Models for Event driven Systems. *Proc. of RTSCA 99*, pp.182-189, IEEE CS, 1999.
8. E. Fersman, P. Pettersson, Wang Yi. Timed Automata with Asynchronous Processes: Schedulability and Decidability. *LNCS 2280*, pp.67-82, 2002.
9. J. McManis, P. Varaiya. Suspension automata: A decidable class of hybrid automata. *LNCS 818*, pp.105-117, 1994.
10. J. C. Corbett. Modeling and analysis of real-time Ada tasking programs. *Proc. of RTSS*, pp. 132-141, IEEE CS, 1994.
11. F. Cassez, F. Laroussinie. Model-checking for hybrid systems by quotienting and constraints solving. *LNCS 1855*, pp. 373-388, 2000.
12. R. Alur, C. Coucoubetis, N. Halbwachs, et-al. The algorithmic analysis of hybrid systems. *TCS*, Vol.138, pp.3-34, 1995.
13. P. Bouyer, C. Dufourd, E. Fleury, and A. Petit. Expressiveness of updatable timed automata. *LNCS 1893*, pp.232-242, 2000.
14. E. D. Jensen, C. D. Locke and H. Tokuda. A time-driven scheduling model for realtime operating systems. *Proc. of RTSS*, pp.112-122, 1985.
15. B. Kao, H. Garcia-Molina, and B. Adelberg. On building distributed soft real-time systems. *Proc. of WPDRTS*, pp.13-19, 1995.
16. K.M. Kavi, H.Y. Youn, B. Shirazi, A.R. Hurson. A Performability Model for Soft Real-Time Systems. *Proc. of HICSS*, pp.571-579, 1994.
17. M. K. Gardner. Probabilistic Analysis and Scheduling of Critical Soft Real-time Systems. *PhD thesis, University of Illinois, Urbana, Illinois*, 1999.
18. H. Hansson and B. Jonsson. A calculus for communicating systems with time and probabilities. *Proc. of RTSS*, pp.278-287, 1990.
19. R. Alur, C. Courcoubetis, D.L. Dill. Model-checking for probabilistic real-time systems. *LNCS 510*, pp. 115-136, 1991.
20. M. Kwiatkowska, G. Norman, R. Segala and J. Sproston. Automatic Verification of Real-Time Systems With Discrete Probability Distributions. *LNCS 1601*, pp. 75-95, 1999
21. P.R. D'Argenio, J.-P. Katoen, E. Brinksma. Specification and Analysis of Soft RealTime Systems: Quantity and Quality. *Proc. of RTSS*, pp. 104-114, 1999.
22. K. Tindell. Fixed-Priority Scheduling of Hard Real-Time Systems. *PhD thesis, University of York, UK*, 1994.
23. D.L. Dill. Timing assumptions and verification of finite-state concurrent systems. *LNCS 407*, pp.197-212, 1989.
24. S. Yamane, K. Nakamura. Symbolic Model-Checking Method Based on Approximations and BDDs for Real-Time Systems. *LNCS 1281*, pp.562-582, 1997.



# A Component-Based Architecture for an Autonomic Middleware Enabling Mobile Access to Grid Infrastructure

Ali Sajjad, Hassan Jameel, Umar Kalim, Young-Koo Lee, and Sungyoung Lee

Department of Computer Engineering, Kyung Hee University, Giheung-Eup,  
Yongin-Si, Gyeonggi-Do, 449-701, Republic of Korea  
{ali, hassan, umar, yklee, sylee}@oslab.khu.ac.kr

**Abstract.** The increasing pervasiveness of wide-area distributed computing resources, like computational Grids, has given rise to applications that have inherent problems of complexity, adaptability, dynamism and heterogeneity etc. The emerging concept of autonomic computing holds the key to the self-management of such a multifarious undertaking and provides a way to further build upon this complexity without incurring additional drawbacks. Furthermore, access to Grid services at present is generally limited to devices having substantial computing, network and memory resources whereas most of mobile devices do not have the sufficient capabilities to be either direct clients or services in the Grid environment. The existing middleware platforms like Globus do not fully address mobility, yet extending the potential of the Grid to a wider audience promises increase in its flexibility and productivity. In this paper<sup>1</sup>, we present a component-based autonomic middleware that can handle the complexity of extending the potential of the Grid to a wider mobile audience, by incorporating the features of context-awareness and self-management. We also address the middleware issues of job delegation to a Grid service, support for disconnected operation/offline processing and secure communication.

## 1 Introduction

Grid [1] computing permits participating entities connected via networks to dynamically share their resources. Its increasing usage and popularity in the scientific community and the prospect of seamless integration and interaction with heterogeneous devices and services makes it possible to develop further complex and dynamic applications for the Grid. However, most of the conventional distributed applications are developed with the assumption that the end-systems possess sufficient resources for the task at hand and the communication infrastructure is relatively reliable. For the same reason, the middleware technologies for such distributed systems encourage the developers to focus on the functionality rather than the distribution. But we know that

---

<sup>1</sup> This research work has been supported in part by the ITRC program of Korean Ministry of Information and Communication (MIC), in collaboration with Sunmoon University, Republic of Korea.

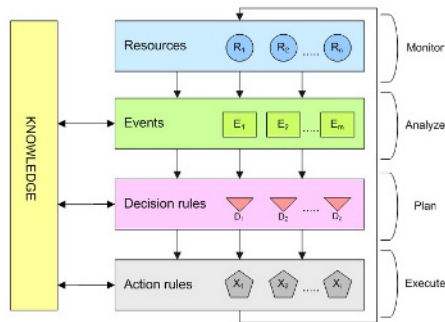
in case of mobile computing, these assumptions are not essentially true. Firstly, there is a wide variety of mobile devices available; laptops offering substantial computing power and memory etc. to cell phones with scarce resources. Secondly, in mobile systems, network connections generally have limited bandwidth, high error rates and frequent disconnections. Lastly, mobile clients usually have the ability to interact with various networks, services, and security matters as they move from one place to another. So extending this potential of the Grid to a wider audience promises increases in flexibility and productivity, particularly for the users of mobile devices who are the prospective consumers and beneficiaries of this technology.

This goal was the main motivation behind the MAGI middleware [25], whose architecture provided the foundation support and infrastructure for building applications and services that provide mobile clients access to Grid. However, the efficient management of such a large platform is a considerably complicated issue and it is a constantly increasing complexity because of increasing numbers of heterogeneous devices and components being added to it. In fact, the current level of software complexity has reached such a level of complexity that it threatens the future growth and benefits of the general IT infrastructure [2]. Also, as the environment of a mobile device changes, the application/service behaviour needs to be adjusted to adapt itself to the new environment. Hence dynamic reconfiguration is an important building block of such an adaptive system. Furthermore, the interaction approach between the mobile client and the host dictates the effectiveness and efficiency of a mobile system. A promising approach to handle this complexity is the emerging field of autonomic computing [3]. It calls for the design of a system in such a way that it is aware of its constituting components and their details, it can reconfigure itself dynamically according to the change in its environment, optimize its working to achieve its goals and predict or recognize faults and problems in its work flow and rectify them. The inspiration of such self-managing approach has been taken from the autonomous nervous system and many efforts are underway for further development in this field [4], [5], [6]. The effect of such an autonomous system will be the reduction in the complexity and ease in management of a large system, and what better exemplar case for its application than the Grid and its middlewares. Various ways have been proposed to achieve the fulfillment of this vision, from agent-based [7] to policy-based self-management [8], from autonomic elements and closed control loops [9] to adaptive systems, self-stabilizing systems and many more, but the motivation and ultimate goal of all is the same. Hence, given the highly variable computing environment of mobile systems, it is mandatory that modern middleware systems are designed in such a way that they can support the requirements of modern mobile systems such as dynamic reconfiguration and asynchronous communication. The motivation behind the AutoMAGI middleware is to develop an autonomic middleware that provides mobile devices access to Grid infrastructure and also enables autonomous applications to use its platform. It will be beneficial to all kinds of Grid users, from the physicist who wants to run a set of simulations from his PDA to a doctor who wants a Grid medical service to analyze the MRI or CT scans of a patient from his smart phone, so that finally we can promise increase in seamless flexibility and productivity. In what follows, we first discuss the basic structure of autonomic components in our autonomic MAGI middleware and then present its architecture, which enables heterogeneous mobile devices to access Grid services and also enables autonomous applications to

use it as a platform for this purpose. This middleware provides support and management infrastructure for delegation of jobs to the Grid, a light-weight security model, offline processing, adaptation to network connectivity issues (disconnected operation) and presentation of results to client devices in keeping with their limited resources.

## 2 Autonomic Components in AutoMAGI

The AutoMAGI middleware is composed of autonomic components which are responsible for managing their own behavior in keeping with the relevant policies, and also for cooperating with other autonomic components to achieve their objectives. The structure of a typical component is shown in Figure 1.



**Fig. 1.** Structure of an Autonomic Component in AutoMAGI

A resource can be anything, from a CPU, memory etc. to an application or service. The event signifies a change in the state of a resource. Decision rules are used for deducing a problem based on the information received from various events. The problems can be deduced using a single event or inferring from a group of events, based on a single occurrence or based on a history of occurrences. The component then makes plans to rectify this problem or optimize some functional/behavioral aspects, basing upon the policies and internal and external knowledge of the component. Action rules are then used to execute tasks to bring about these changes in line with the desired goal of the component.

This manner of structure facilitates in building up a control loop by employing the monitor, analyze, plan and execute cycle, which is of key significance for autonomic behavior [24].

## 3 AutoMAGI Architecture

The AutoMAGI middleware is exposed as a web service to the client application. The components of the middleware (as shown in Figure 2) are discussed briefly as follows.

### 3.1 Discovery Service

The discovery of the middleware by mobile devices is undertaken by employing a UDDI registry [11], [12]. The composition of the current web services may not give sufficient facilities to depict an autonomic behavior or to integrate them seamlessly with other autonomic components but with the advent of semantic web service technologies like OWL-S [13], it becomes possible to provide a fundamental framework for representing and relating devices and services with their policies and describing and reasoning about their functionalities and capabilities. Another hurdle is that the current organization and management of Web services and Grid services are static and must be defined a priori. However, by using Web Services Distributed Management (WSDM) [26], we get a mechanism of managing resource-oriented and dynamic web services and their discovery.

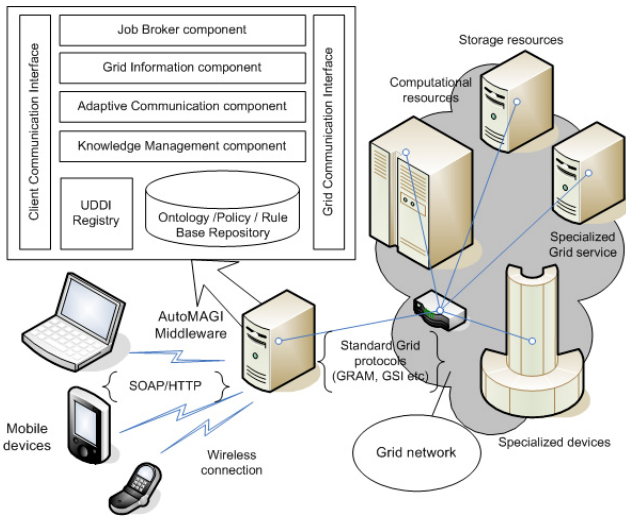


Fig. 2. AutoMAGI – Architecture and Deployment diagram

Once the middleware service is deployed and registered, other applications/devices would be able to discover and invoke it using the API in the UDDI specification [12] which is defined in XML, wrapped in a SOAP envelope and sent over HTTP.

### 3.2 Client Communication Interface

The service advertised to the client is the communication interface between the mobile device and the middleware. This layer enables the middleware to operate as a semantic web service [14] and communicate via the SOAP framework [15]. The repository in the middleware also contains the device and user related ontologies and service policies for the devices, users and applications. Due to this service-oriented approach, it is not expected of the client to remain connected to the middleware at all times while the request is being processed.



protocols such as GRAM [16], MDS [17], GSI [18] etc. which are obligatory for any application trying to communicate with the Grid services using Globus. This enables the middleware to communicate with the Grid, in order to accomplish the job assigned by the client. Its working is in close collaboration with the Grid Information component, which is discussed later in the section.

### 3.5 Grid Information Component

The Grid Information component interacts with the wrapper of the GLOBUS toolkit's API for information services (MDS [17]). It assists the client application by managing the process of determining which services and resources are available in the Grid (the description of the services as well as resource monitoring such as CPU load, free memory etc.). Detailed information about Grid nodes (which is made available by MDS) is also shared on explicit request by the client.

### 3.6 Job Broker Component

The autonomic Job Broker component deals with initiating the job request and steering it on behalf of the client application. After going through the related policy and determining the availability of the Grid service and authorization of the client, it downloads the code (from the mobile device or from a location specified by the client e.g. an FTP/web server). Once the code is available, the Job Broker component submits a "createService" request on the GRAM's Master Managed Job Factory Service (via the wrapper) which is received by the Redirector [16]. The application code (controlled by the application policy) then interacts with the newly created instance of the service to accomplish the task. The rest of the process including creating a Virtual Host Environment (VHE) process and submitting the job to a scheduling system is done by GRAM. Subsequent requests by the client code to the Job Broker component are redirected through the GRAM's Redirector.

The monitoring service of the Job Broker component interacts with GRAM's wrapper to submit FindServiceData requests in order to determine the status of the job. It may then communicate with the Knowledge Management component to store the results, depending on the type of application and all the related policies, as the mobile client may reconnect and ask for the results (intermediate/final) of its job from the Job Broker component.

### 3.7 Knowledge Management Component

The knowledge used by different autonomic components is handled by using semantic web technologies in the middleware which provide the mechanisms to present the information as machine-processable semantics and is useful in building intelligent decision-making mechanisms and perform knowledge level transformations on that information. These decisions and transformed information is then passed on to other components within the system or directly to the client or the Grid, which utilize it according to their specific needs.

The autonomic components that constitute the AutoMAGI middleware constantly monitor and gather the data they need to react to or act upon, according to their management tasks and targets. This wide-scoped data is elaborated and organized through

the notion of events. Events in turn are typically meaningful in a certain context when related with other events. This correlation information can then be used for Data filtering, measuring thresholds and sequencing of actions etc. But for such an autonomic model to work properly, a shared knowledge must be present which includes features context information, system logs, performance metrics and relevant policies. We manage this knowledge base with the help of a Policy Manager in the KM component. Due to the autonomic character of the Knowledge Management component, the middleware is able to respond to a problem that happened in the defined problem space (scope of defined problems) and use predictive methods to discover probable problems in advance and so succeed in achieving better results and eliminating problems. But as each autonomic element has its own knowledge model, the problem of data/knowledge integration might result, which is again handled by the Knowledge Management module. Furthermore, some conflict-scenarios may arise due to the conflicting goals pursued by different autonomic components. The optimal solution of such conflicts is also the job of this component.

### 3.8 Security Component

The Grid Security Infrastructure is based on public key scheme mainly deployed using the RSA algorithm [19]. However key sizes in the RSA scheme are large and thus computationally heavy on handheld devices such as PDA's, mobile phone's, smart phones etc. We propose the use of Elliptic Curve Cryptography (ECC) based public key scheme, which can be used in conjunction with Advanced Encryption Standard (AES) for mobile access to Grid. This provides the same level of security as RSA and yet the key sizes are a lot smaller [20] which means faster computation, low memory and bandwidth and power consumption with high level of security.

Furthermore, as no autonomic element should provide its resources or services to any other component without the permission of its manager, we make use of security policies that govern and constrain their behavioral aspects at a higher level. The security policies include different characteristics like the level of protection needed to be applied to the various information resources that the component contains or controls, rules that determine how much trust the element places in other elements with which it communicates, cryptographic protocols the element should use in various situations and the circumstances in which the element should apply or accept security-related patches or other updates to its own software, and so on. Each autonomic component also holds various security related tasks and state representations to describe the current status and activities, like level of trust on other communicating entities, notification form other components or human administrators of suspicious circumstances, agreements with other components regarding provision of security-related information, such as log-file analyses or secure time stamping, and a list of trustworthy resource suppliers (used to quickly verify the digital signatures on the resources they provide).

## 4 Communication Between the Middleware Gateways

In case multiple instances of the MAGI middleware gateways are introduced for improving scalability, some problem scenarios might arise. Consider a mobile device that accesses the Grid network via gateway  $M_1$ , but disconnects after submitting the job. If

the mobile device later reconnects at gateway  $M_2$  and inquires about its job status, the system would be unable to respond if the middleware is not capable of sharing information with other instances. To manage resources, clients and requests etc. between themselves, the distributed instances of AutoMAGI middleware use an Arbiter component. So in accordance with high-level guidance from the application/client's policies for the functional environment and the load-balancing policies from the middleware, we attain a guideline for optimal sharing of knowledge between different middleware instances.

The Arbitrator facilitates in communication between any two middleware instances. It maintains the ordered pairs (ID, URI) which are used for the identification of the middleware instance. So for instance, after reintegration of the mobile client at  $M_2$ ,  $C$  sends the ID of the middleware instance, where the job was submitted (i.e.  $M_1$ ), to the Arbitrator. The Arbitrator determines that the ID is not that of  $M_2$ . It then checks the Middleware Directory Listing to find the URI corresponding to the Middleware instance  $M_1$ . The Arbitrator then requests (from the client application) the job-ID of the job submitted by  $C$ . Upon a successful response the Arbitrator of  $M_2$  ( $A-M_2$ ) communicates with the Arbitrator of  $M_1$  ( $A-M_1$ ) using the URI retrieved. After mutual authentication,  $A-M_2$  sends the job-ID along with the clients request for fetching the (intermediate/final) results to  $A-M_1$ . If the job is complete, the compiled results are forwarded to client application. In case the job isn't complete yet, the client application continues to interact with  $A-M_1$  (where the job was submitted). Note that  $A-M_2$  acts as a broker for communication between  $C$  and  $M_1$ . Also, if the  $C$  decides to disconnect and later reconnect at a third middleware instance  $M_3$ , then  $A-M_3$  will act as a broker and communicate with  $M_1$  on behalf of  $C$ . As all the processing of information is done at the middleware where the job was submitted, the other instances would only act as message forwarding agents.

## 5 Related Work

Signal [21] proposes a mobile proxy-based architecture that can execute jobs submitted to mobile devices, so in-effect making a grid of mobile devices. After the proxy server determines resource availability, the adaptation middleware layer component in the server sends the job request to remote locations. The efforts are inclined towards QoS issues such as management of allocated resources, support for QoS guarantees at application, middleware and network layer and support of resource and service discoveries based on QoS properties.

In [22] a mobile agent paradigm is used to develop a middleware to allow mobile users' access to the Grid and it focuses on providing this access transparently and keeping the mobile host connected to the service. Though improvement is needed in the system's security, fault-tolerance and QoS, the architecture is sufficiently scalable. GridBlocks [23] builds a Grid application framework with standardized interfaces facilitating the creation of end user services. For security, they are inclined towards the MIDP specification version 2 which includes security features on Transport layer. They advocate the use of propriety communication protocols based on the statement that performance of SOAP on mobile devices is 2-3 times slower as compared to a proprietary protocol. But in our view, proprietary interfaces limit interoperability and extensibility, especially to new platforms such as personal mobile devices and certainly an autonomic computing system will only be possible if open standards are ensured.



## 6 Conclusions and Future Work

In this paper we identified the potential of enabling mobile devices access to the Grid and how we use the emerging autonomic computing paradigm to solve the management complexity. The component-based architecture of an autonomic middleware named AutoMAGI is presented which facilitates implicit interaction of mobile devices with Grid infrastructure. It ensures secure communication between the client and the middleware service, provides support for offline processing, manages the presentation of results to heterogeneous devices considering the device specifications and deals with the delegation of job requests from the client to the Grid.

In future we intend to focus on issues of autonomic security (self-protection) and streamline the Knowledge Management component for self-optimization. Along with a prototype implementation, we intend to continue validating our approach by experimental results and benchmarks.

## References

1. Foster, I., Kesselman, C., Tuecke, S.: The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *Int'l J. Supercomputer Applications*, vol. 15, no. 3 (2001) 200-222
2. Wladawsky-Berger, I.: *Advancing E-business into the Future: The Grid*. Kennedy Consulting Summit. New York (2001)
3. Ganek, A.G., Corbi, T.A.: The dawning of the autonomic computing era. *IBM Systems Journal*, v.42 n.1 (2003) 5-18
4. Horn, P.: *Autonomic Computing: IBM's Perspective on the State of Information Technology*. [http://www.research.ibm.com/autonomic/manifesto/autonomic\\_computing.pdf](http://www.research.ibm.com/autonomic/manifesto/autonomic_computing.pdf): IBM Corporation (2001)
5. HP Adaptive Enterprise strategy: <http://www.hp.com/go/demandmore>
6. Microsoft Dynamic Systems Initiative: <http://www.microsoft.com/windowsserversystem/dsi/default.mspx>
7. Bonino, D., Bosca, A., Corno, F.: *An Agent Based Autonomic Semantic Platform*. First International Conference on Autonomic Computing. New York (2004)
8. Chan, H., Arnold, B.: A policy based system to incorporate self-managing behaviors in applications. Companion of the 18th Annual ACM SIGPLAN Conference on Object-oriented Programming, Systems, Languages, and Applications. (2003) 94-95
9. Kephart, J., Chess, D.: The Vision of Autonomic Computing. *Computer*, v.36 n.1 (2003) 41-50
10. Puliafito, A., Riccobene, S., Scarpa, M.: Which paradigm should I use?: An analytical comparison of the client-server, remote evaluation and mobile agents paradigms'. *IEEE Concurrency and Computation: Practice & Experience*, vol. 13 (2001) 71-94
11. Hoschek, W.: *Web service discovery processing steps*. <http://www-itg.lbl.gov/~hoschek/publications/icwi2002.pdf>
12. UDDI specification: <http://www.oasis-open.org/committees/uddi-spec/doc/tcspecs.htm>
13. OWL-S: OWL-based Web Service Ontology: <http://www.daml.org/services/owl-s/>
14. Semantic Web Services Initiative (SWSI): <http://www.swsi.org/>
15. SOAP Framework: W3C Simple Object Access Protocol ver 1.1, World Wide Web Consortium recommendation. 8 May 2000; [www.w3.org/TR/SOAP/](http://www.w3.org/TR/SOAP/)
16. GT3 GRAM Architecture: [www-unix.globus.org/developer/gram-architecture.html](http://www-unix.globus.org/developer/gram-architecture.html)

17. Czajkowski, K., Fitzgerald, S., Foster, I., Kesselman, I.: Grid Information Services for Distributed Resource Sharing. Proceedings of the Tenth IEEE International Symposium on High-Performance Distributed Computing, IEEE Press (2001)
18. Welch, V., Siebenlist, F., Foster I., et al.: Security for Grid Services. HPDC (2003)
19. Welch, V., Foster I., Kesselman, C., et al.: X.509 Proxy Certificates for dynamic delegation. Proceedings of the 3rd Annual PKI R&D Workshop (2004)
20. Gupta, V., Gupta, S., et al.: Performance Analysis of Elliptic Curve Cryptography for SSL. Proceedings of ACM Workshop on Wireless Security. Atlanta, GA, USA (2002) 87-94
21. Hwang, J., Aravamudham, P.: Middleware Services for P2P Computing in Wireless Grid Networks. IEEE Internet Computing vol. 8, no. 4 (2004) 40-46
22. Bruneo, D., Scarpa, M., Zaia, A., Puliafito, A.: Communication Paradigms for Mobile Grid Users. Proceedings 10th IEEE International Symposium in High-Performance Distributed Computing (2001)
23. GridBlocks: Helsinki Institute of Physics, (CERN).  
<http://gridblocks.sourceforge.net/docs.htm>
24. Herrmann, K., Mühl, G., Geihs, K.: Self-Management: The Solution to Complexity or Just Another Problem? IEEE Distributed Systems Online, vol. 6, no. 1 (2005)
25. Sajjad, A., Jameel, H., et al.: MAGI - Mobile Access to Grid Infrastructure: Bringing the gifts of Grid to Mobile Computing. Proceedings of 2nd International Conference on Grid Service Engineering and Management. Erfurt, Germany (2005)
26. Web Services Distributed Management (WSDM) standards specifications:  
<http://docs.oasis-open.org/wsdm/2004/12/wsdm-1.0.zip/>

# Autonomic Agents for Survivable Security Systems

Roy Sterritt, Grainne Garrity, Edward Hanna, and Patricia O'Hagan

<sup>1</sup> University of Ulster, School of Computing and Mathematics,  
Jordanstown Campus, Northern Ireland

R.Sterritt@ulster.ac.uk, Grainne@coresystems.biz

<sup>2</sup> Core Systems, Belfast, Northern Ireland

Edward@coresystems.biz, Patricia@coresystems.biz

**Abstract.** Autonomic Systems are essentially about creating self-managing systems based on the biological metaphor of the non-conscious acting autonomic nervous system. The Autonomic initiative has been motivated by ever increasing complexity and total cost of ownership of today's system of systems. Autonomicity also offers inroads in terms of fault-tolerant computing and assisting in creating survivable systems. This paper examines the relevant technologies including Agents for engineering autonomicity and survivability in a secure location biometric system.

## 1 Introduction

The case has been well presented in the literature for the need to create self-managing systems due to the complexity problem, the total cost of ownership, or to provide the way forward to enable future pervasive and ubiquitous computation and communications [1][2][3][4]. Another aspect for self-management is to facilitate survivable systems. To enable self-management (*autonomicity*) a system requires many *self* properties (*self*-\* or *selfware*), such as self-awareness. This paper first looks at autonomic management, it then describes a deployed critical security system that requires built in survivability and autonomicity. The paper then briefly examines related work, namely agents and clusters before going on to present an agent based approach to deal with providing self-management redundancy in the system and avoid the traditional updating issues with failover servers.

## 2 Autonomic Management

Computing systems are expected to be *effective*. This means that they serve a useful purpose when they are first introduced and continue to be useful as conditions change. Responses taken automatically by a system without real-time human intervention are autonomic responses [5]. The autonomic concept is inspired by the human body's autonomic nervous system. By analogy, humans have good mechanisms for adapting to changing environments and repairing minor physical damage. The autonomic nervous system monitors heartbeat, checks blood sugar levels and keeps the body temperature normal without any conscious effort from the human. This biological autonomicity is influencing a new paradigm for computing to create similar self-

management within systems (Autonomic Computing, Autonomic Communications and Autonomic Systems). There is an important distinction between autonomic activity in the human body and autonomic responses in computer systems. Many of the decisions made by autonomic elements in the body are involuntary, whereas autonomic elements in computer systems make decisions based on tasks chosen to delegate to the technology [6].

Many branches of computer science research and development will contribute to progress in autonomic computing. In particular, it brings together work in software engineering and artificial intelligence. Research on dependable systems should be especially influential, as dependability covers many relevant system properties such as *reliability*, *availability*, *safety*, *security*, *survivability* and *maintainability* [7],[8].

In the late 1990s DARPA/ISO's Autonomic Information Assurance (AIA) programme studied defence mechanisms for information systems against malicious adversaries. The AIA programme resulted in two hypotheses; (1) fast responses are necessary to counter advance cyber-adversaries and (2) coordinated responses are more effective than local reactive responses [5]. These hypotheses may provide general guidance for creating autonomic survivable systems.

'Autonomic' became mainstream within Computing in 2001 when IBM launched their perspective on the state of information technology [1]. IBM defined four key self properties: *self-configuring*, *self-healing*, *self-optimizing* and *self-protecting* [6]. In the few years since, the *self-x* list has grown as research expands, bringing about the general term *selfware* or *self-\**, yet these four initial self-managing properties along with the four enabling properties; *self-aware (of internal capabilities and state of the managed component)*, *self-situated (environment and context awareness)*, *self-monitor* and *self-adjust (through sensors, effectors and control loops)*, cover the general goal of self management [8].

The influence of the autonomic nervous system (ANS) may imply that the Autonomic Computing initiative is concerned only with low level self-managing capabilities such as reflex reactions. This fits with AIA perspective, other layered architectures such as; reaction—routine—reflection [9]; data—management/control—knowledge planes [10]; hardware—cyber—mission planes [5]; autonomic—selfware—autonomous [11]. Yet within the autonomic research community the vision behind the initiative is an overarching goal of system-wide policy-based self-management where a human manager will state a business-critical success factor and the ICT systems will take care of it, self-configuring and self-optimising to meet the policies, and self-protecting and self-healing to ensure the policies are maintained in light of changing circumstances. It may be reasoned that due to our ANS we are freed (non-conscious activity) from the low-level complexity of managing our bodies to perform high-level complex tasks. Similarly, for Computing to develop further and provide equivalent high-level system-wide tasks, necessitates a corresponding low-level 'non-conscious' architecture. As such, increasing this initiative will converge and cross-influence the fields of ubiquitous and pervasive computing.

### 3 The Core Survivable Security System

Highly secure locations, such as research labs and law enforcement correction centers, are seeking to utilize the latest technology in biometrics such as iris and finger print

technologies to allow and restrict access and movement around their locations [12]. Architectural requirements for such systems dictate that the architecture should be secure, fault-tolerant, and non-exploitable [5]; that the system must meet the security policies of the organization and it must accommodate different security infrastructures; the system must remain robust and secure when faults occur, both random faults caused by the failure of system elements and to malicious faults caused by a deliberate attack on the system; that communication must be reliable so that defensive components remain fully functional even in the face of an attack on the infrastructure; and that it must not be possible for an attacker to exploit defensive components to effect an undesirable action, for example, an attacker or random fault must not be able to trigger a response that causes the system to unnecessarily deny legitimate users access and movement nor the opposite allow non-legitimate users access and movement around the location [5].

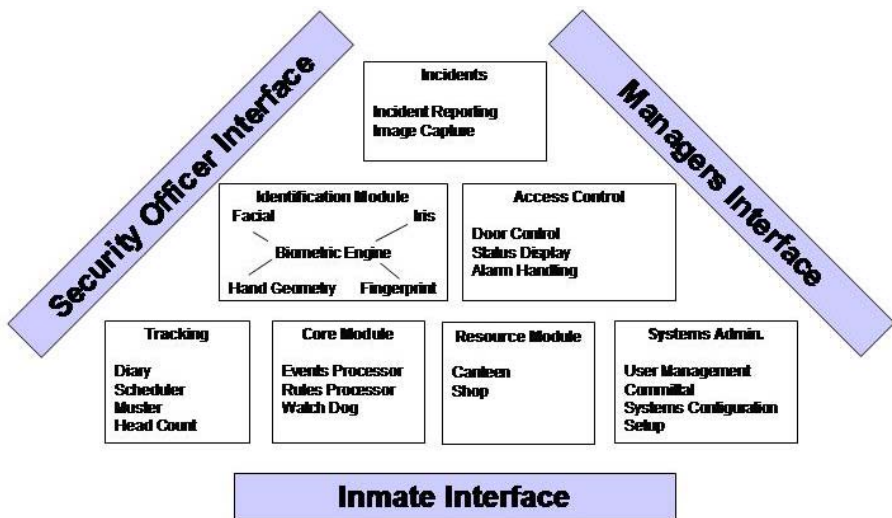


Fig. 1. Core Security System - High Level Architecture

Figure 1 depicts a high level architecture for access control to a correction center secure location incorporating such identification technologies as biometrics [12]. The implication of these multiple components is that the system becomes more complex, a theme the Autonomic initiatives aim to address. In this situation the self-healing and self-protections aspects are more critical to the system. The continuous monitoring of the system health through control loops matches the architecture needs. Research within the Autonomic field such as self-healing and micro rebooting can assist in creating a survivable environment [8]-[14].

Figure 2 illustrates the Survivable Secure System (SSS) architecture. It provides for an inmate tracking solution that delivers the information needed to manage the correction center. The system supports cell assignments and tracking of inmates' movements, various activities and events. The SSS is a modular system that will integrate with existing systems with the general concept being to monitor people

within a defined space and manage their movements, access to goods, services and privileges. A predefined set of rules can be applied at any level to limit movement, interaction between individuals or access to resources.

A history will build up of activity for each individual defined to the system. Biometric technology is used to uniquely identify prisoners. This makes it possible to provide inmates with personal information and enable them to request services through the system.

### 3.1 Survivability

Survivability is achieved within the system through the use of Watchdog, heartbeats, micro-reboots, automated failover and a dual redundant ring architecture.

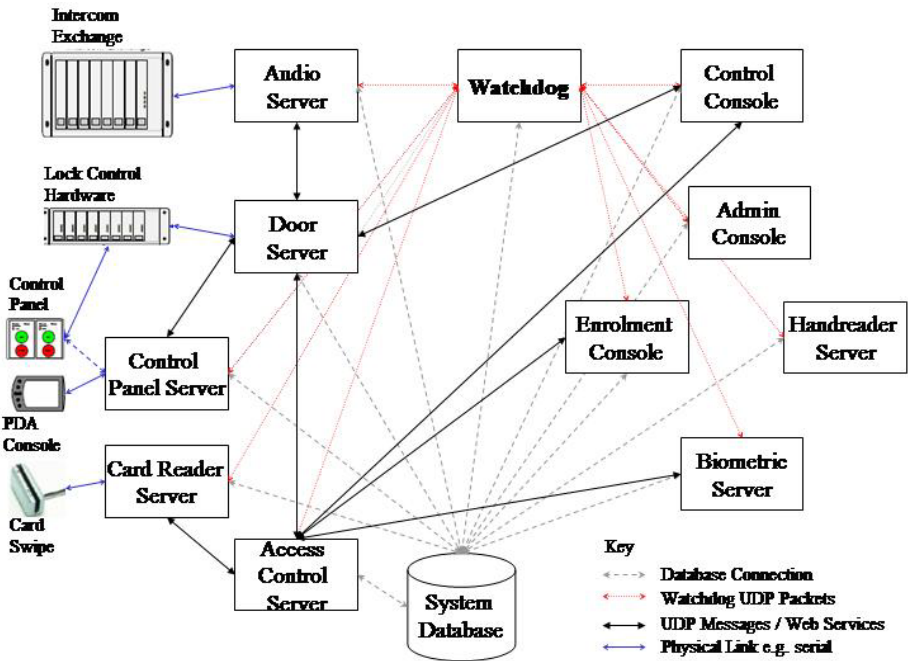


Fig. 2. Core Security System - Component Architecture

#### 3.1.1 Watchdog

Watchdog is essentially status monitoring software, providing a service that monitors the status of a specific list of devices/programs across the network. This status information is sent to registered consoles where it is displayed in a graphical format representing the system. From the graphic layout users are shown the current system status and alerted to any problems. This enables the user to initiate pre-programmed remedial action. It is also possible to have the Watchdog perform automatic responses to specific problems.

The Watchdog communicates with hardware devices either using standard pings or by using SNMP (Simple Network Management Protocol) messages. By using a standard ping the Watchdog can only ascertain if the device is replying or not. By using SNMP the Watchdog can collect more detailed information about the device, in the case of SNMP enabled switches (e.g. Netgear) individual links can be monitored. For software applications, the Watchdog sends heartbeat (Are you alive?) messages to which the application replies giving its status. The heartbeat messages also contain the status of other applications which have a functional relation to the application being polled. This additional information can be used by the application to make decisions on how to operate if a required function of the system is not working correctly.

If a problem is detected in an application the Watchdog can kill the faulty application and restart it. If a problem is detected in hardware the Watchdog can restart the device via software controlled power management.

### **3.1.2 Heart-Beats**

Essentially, one view of the goal of autonomic computing is to create robust dependable self-managing systems [8]. To facilitate this aim, fault-tolerant mechanisms such as a heart-beat monitor ('I am alive' signals) and pulse monitor (urgency/reflex signals) may be included within the autonomic element [9][2]. The notion behind the pulse monitor (PBM) is to provide an early warning of a condition so that preparations can be made to handle the processing load of diagnosis and planning a response, including diversion of load. Together with other forms of communications it creates dynamics of autonomic responses [15] – the introduction of multiple loops of control, some slow and precise, others fast and possibly imprecise, fitting with the biological metaphor of reflex and healing [9].

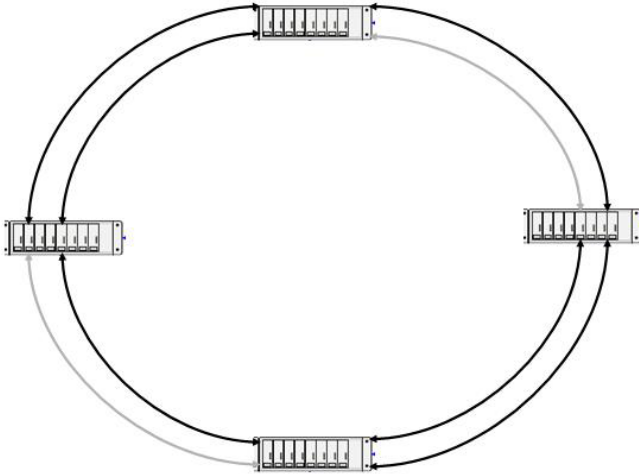
### **3.1.3 Micro-rebooting**

One promising avenue of research for self-healing (in particular in software) is microrebooting [13]. In this approach components at varying granularity levels are designed to be crash only, that is if they are not functioning correctly, self-healing is simply a reboot.

Since rebooting can be expensive causing non trivial service disruption or downtime (even when failover and clusters are employed) the key is to separate process recovery from data recovery and to achieve fine grained rebooting; i.e. components as opposed to applications or even systems [13].

### **3.1.4 Redundant Ring Architecture**

Survivability is achieved within the network communications through a dual redundant fibre ring architecture. This is achieved by using Netgear layer 2 managed switches. For an Ethernet network to function properly, only one active path can exist between two stations. Multiple active paths between stations cause loops in the network. If a loop exists in the network topology, the potential exists for duplication of messages. To provide path redundancy on Ethernet networks the Spanning-Tree Protocol can be used. Spanning-Tree Protocol forces certain redundant data paths into a standby (blocked) state. If one network segment in the Spanning-Tree Protocol becomes unreachable, the spanning-tree algorithm reconfigures the spanning-tree topology and re-establishes the link by activating the standby path. This reconfiguration mechanism is transparent to the machines on the network. The blocked standby paths are illustrated as lighter grey connection in Figure 3.



**Fig. 3.** Dual Redundant Ring Architecture

### 3.1.5 Failover/Load Balancing

The major server applications, illustrated in Figure 2 exist as multiple instances running on different machines. The load is balanced between these instances so that in the event of a machine failure there is minimal impact on the system. Local fail safe has been built into the access control system so that under extreme fault conditions the access control system will still function in a limited capacity.

## 4 Related Work

Autonomic Computing is dependent on many disciplines for its success; not least of these is research in agent technologies. At this stage, there are no assumptions that agents have to be used in an autonomic architecture, yet the desired self-\* properties can benefit from the nature of agents, namely; autonomy, reactivity, sociality and pro-activity. Also, as in complex systems there are arguments for designing the system with agents [16], as well as providing inbuilt redundancy and greater robustness [17], through to retrofitting legacy systems with autonomic capabilities that may benefit from an agent approach [18].

Emerging research ranges from autonomic architecture containing autonomic managers as agents themselves, a self-managing cell [19], containing functionality for measurement and event correlation and support for policy-based control, to alternative rational models, such as state-feeling-action [20], that may better suit autonomic systems than traditional DBI models. Autonomics is also finding its way into agent research, for instance semantic web agents [21].

Cluster computing, whereby a large number of simple processors or nodes are combined together to apparently function as a single powerful computer, has emerged as a research area in its own right. Traditionally, Massively Parallel Processing (MPP) computer systems have been used to meet high performance computing requirements. MPP computers may contain hundreds or thousands of processors within a single computer system. Typically, upgrading such systems requires a



complete rebuild of the system. They are, however, relatively simple to manage, and they certainly perform very well. A recent trend in high performance computing research has been to find new approaches to overcome the cost and scalability issues associated with MPP systems, such as clusters and grids.

The Cluster approach offers a relatively inexpensive means of providing a fault-tolerant environment and achieving significant computational capabilities for high-performance computing applications. However, the task of manually managing and configuring a cluster quickly becomes daunting as the cluster grows in size. Autonomic computing is also being researched to provide self-managing cluster systems, for instance the Autonomic Cluster Management System (ACMS) [22] that exploits autonomic properties and agents in automating cluster management.

## 5 Autonomic Management with Agents

A major issue with providing redundancy and automated failover is the expense in keeping the back-up servers up-to-date. When you also consider the SSS environment where there are many types of servers this becomes particularly critical. If one utilizes autonomic agents to design the self-management system, then such properties as agent cloning may be utilized to provide up-to-date back-up redundant servers.

This section considers the autonomic self-managing survivable system as a mobile agent system, much the same as the ACMS [22]. The system is composed of a number of agent processes, representing both the required servers (control panel server, door server, audio server and so on) and the necessary autonomic management agents, communicating across a network of nodes. The system consists of four types of agents, each with functionality implementing autonomic system properties, namely the *General Server Agents (GA)*, *Health Agents (HA)*, *Optimization Agents (OA)*, and *Configuration Agents (CA)*.

One configuration to provide the survivable autonomicity is that the system is comprised of two Configuration Agents, two Health Agents, and one Optimization Agent per implementation, as well as two General Agents per node (primary and redundant secondary). Each general agent is designed to be specific-purpose, and to perform a particular task (the servers in the SSS). The community of agents collaborates to achieve a common goal, specifically providing autonomic management of the system, while simultaneously maximizing performance by implementing load-balancing techniques on the system. Figure 4 illustrates the architecture of the proposed system.

The agents function includes a heartbeat (sending of a periodic 'I am alive' signal) [9], to facilitate fault tolerance and the provision of localized fail-over on the node; i.e., instead of the traditional approach of the CA noticing an agent has failed (through polling) on a remote host and instructing the switch over to the secondary agent on that remote host, this can occur locally via the secondary agent monitoring the heartbeat from the primary agent and thus providing a tighter and situated reflex reaction upon failure.

The primary and secondary CAs will also utilize the same mechanism with heartbeats between them (typically on separate hosts to increase fault tolerance). Self-healing is provided for explicitly through a health agent. Its function, in collaboration with the CAs and OAs, will be to monitor vital signs on the hosts in an attempt to predict if a host is having difficulties and a failure is imminent.

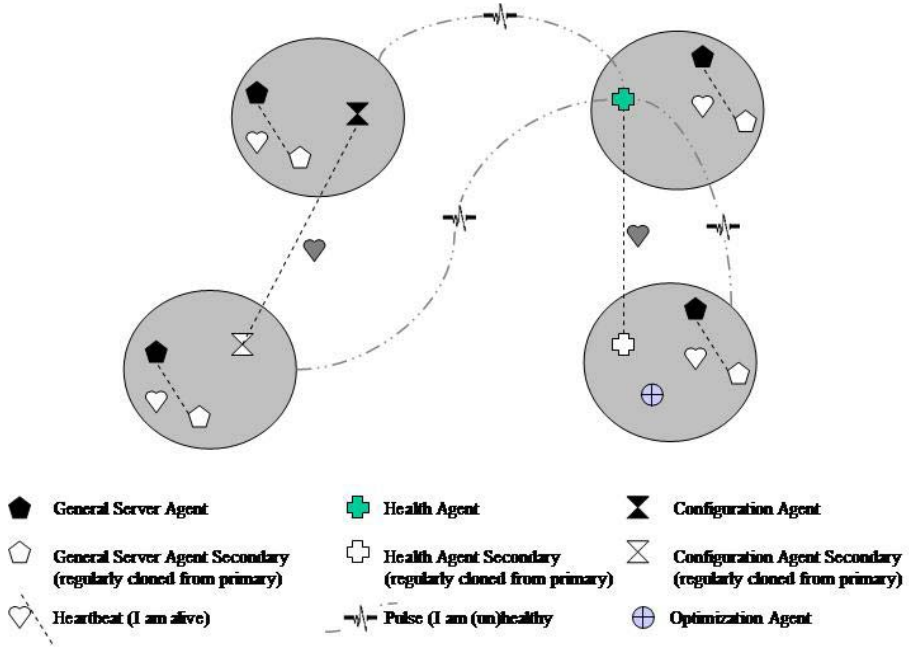


Fig. 4. Autonomic Agents Survivable Self-Management System

The health agent will be facilitated by pulse monitoring – the extension of heartbeat monitoring from ‘I am alive’ signals to include health information within the signal, akin to measuring the pulse [9]. In this scenario the local agents may failover with a new secondary agent being created, yet if this starts to occur frequently this may indicate the host itself is unstable. The health agent can monitor developing scenarios and work with the configuration agent to avoid allocating work to unstable hosts [22].

## 6 Conclusion

This paper has highlighted that Autonomic Systems are not just motivated by concerns over complexity and total cost of ownership, but also extending fault-tolerance to create survivable systems by utilizing self-\* properties.

We have described a deployed Biometric Identification and Tracking System incorporating survivability and autonomicity, used in law enforcement and correction centers with the aim to monitor people within the establishment in terms of movements, interactions and access to resources.

In these environments it is simply not acceptable for failure at the system level and as such the architecture must have multiple mechanisms to ensure survivability. In the development of the architecture it has been endeavored to incorporate into its design the AIA hypotheses of fast and coordinated responses. The paper also considered a proposed mobile agent solution to increase the flexibility and survivability within the system.

Future extensions of this work will include adding layers above the survivable autonomic system such as Knowledge-based alerting to incidents occurring with the correction center.

## Acknowledgements

The development of this paper was supported by a Knowledge Transfer Partnership project funded by the UK Government's Department of Trade and Industry (DTI). Core Systems' development project is partly supported by InvestNI. The wider context of the Autonomic Systems research is supported at University of Ulster by the Computer Science Research Institute (CSRI) and the Centre for Software Process Technologies (CSPT), funded by Invest NI through the Centres of Excellence Programme, under the EU Peace II initiative.

## References

1. P. Horn, "Autonomic computing: IBM perspective on the state of information technology," IBM T.J. Watson Labs, NY, 15th October 2001.
2. R. Sterritt, "Towards Autonomic Computing: Effective Event Management", Proc. IEEE/NASA SEW, Greenbelt, MD, Dec. 2002.
3. JO Kephart, DM Chess. "The Vision of Autonomic Computing", Computer, 36(1):41–52, 2003.
4. R. Sterritt, "Autonomic Computing", Innovations in Systems and Software Engineering, Vol. 1, No. 1, Springer, pp 79-88, 2005
5. SM Lewandowski, DJ Van Hook, GC O'Leary, JW Haines, LM Rossey, "SARA: Survivable Autonomic Response Architecture", DARPA Information Survivability Conference and Exposition II Proceedings, Vol. 1, pp. 77-88, June 2001.
6. IBM, "An architectural blueprint for autonomic computing", 2003.
7. B. Randell, "Turing Memorial Lecture – Facing Up to Faults", Comp. J. 43(2), pp 95-106, 2000.
8. R Sterritt, DW Bustard, "Autonomic Computing: a Means of Achieving Dependability?", Proc IEEE Int. Conf. on the Engineering of Computer Based Systems (ECBS'03), Huntsville, Alabama, USA, April 7-11 2003, pp 247-251.
9. R. Sterritt, "Pulse Monitoring: Extending the Health-check for the Autonomic GRID", Proceedings of IEEE Workshop on Autonomic Computing Principles and Architectures (AUCOPA 2003) at INDIN 2003, Banff, Alberta, Canada, 22-23 August 2003, pp 433-440.
10. D Clark, C Partridge, JC Ramming, JT Wroclawski, "A Knowledge Plane for the Internet", Proc. Applications, technologies, architectures, and protocols for computer communication, Karlsruhe, ACM SIGCOMM 2003
11. R. Sterritt, MG Hinchey, (Jun 2005) "SPAACE :: Self- Properties for an Autonomous & Autonomic Computing Environment", Proceedings of Autonomic & Autonomous Space Exploration Systems (A&A-SES-1) at 2005 International Conference on Software Engineering Research and Practice (SERP'05), Las Vegas, June 27-30, CSREA Press, Pages 3-8
12. R. Sterritt, G. Garrity, E. Hanna, P. O'Hagan, "Survivable Security Systems through Autonomicity", 2nd NASA GSFC/IEEE Workshop on Radical Agent Concepts (WRAC II), 2005

13. G. Candea, S. Kawamoto, Y. Fujiki, G. Friedman, A. Fox, Microreboot - A Technique for Cheap Recovery. 6th Symp Operating Systems Design and Implementation (OSDI), San Francisco, CA, December 2004
14. R. Sterritt, "Autonomic Networks: Engineering the Self-Healing Property", Engineering Applications of Artificial Intelligence, Vol. 17, No. 7, Elsevier, ISSN 0952-1976, Pages 727-739, Oct 2004
15. R. Sterritt, D.F. Bantz, "PAC-MEN: Personal Autonomic Computing Monitoring Environments," Proc IEEE DEXA 2004 Workshops - 2nd Int. Workshop on Self-Adaptive and Autonomic Computing Systems (SAACS 04), Zaragoza, Spain, Aug. 30 – 3 Sept., 2003.
16. N.R. Jennings, M. Wooldridge, "Agent-oriented Software Engineering," in J. Bradshaw (ed.), Handbook of Agent Technology, AAAI/MIT Press, 2000.
17. M.N. Huhns, V.T. Holderfield, R.L.Z. Gutierrez, "Robust software via agent-based redundancy," Second International Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2003, July 14-18, 2003, Melbourne, Victoria, Australia, pp 1018-1019.
18. G. Kaiser, J. Parekh, P. Gross, G. Valetto, "Kinesthetics eXtreme: An External Infrastructure for Monitoring Distributed Legacy Systems," Autonomic Computing Workshop – IEEE 5th Int. Active Middleware Workshop, Seattle, USA, June 2003.
19. E. Lupu, et al., EPSRC AMUSE: Autonomic Management of Ubiquitous Systems for e-Health, 2003.
20. Hu Jun, Gao Ji, Huang Zhongchao, Liao Beishui, Li Changyun, Chen Jiujun, A New Rational Model for Agent for Autonomic Computing, 2004 IEEE International Conference on Systems, Man and Cybernetics, pp 5531-5536, 2004.
21. V. Tamma, I. Blacoe, B. Lithgow Smith, M. Wooldridge "Introducing autonomic behaviour in semantic web agents", Proceedings of the Fourth International Semantic Web Conference (ISWC), Galway, Ireland, November 2005.
22. WF Truszkowski, MG Hinchey, R Sterritt, "Towards an Autonomic Cluster Management System (ACMS) with Reflex Autonomicity", Proc Workshop Reliability and Autonomic Management in Parallel and Distributed Systems (RAMPDS-05) at ICPADS-2005, Fukuoka, Japan, July 20-22, pp 478-482, July 2005

# Towards Formal Specification and Generation of Autonomic Policies

Roy Sterritt<sup>1</sup>, Michael G. Hinchey<sup>2</sup>, James L. Rash<sup>3</sup>, Walt Truszkowski<sup>3</sup>,  
Christopher A. Rouff<sup>4</sup>, and Denis Gracanin<sup>5</sup>

<sup>1</sup> University of Ulster, Faculty of Engineering, Northern Ireland  
r.sterritt@ulster.ac.uk

<sup>2</sup> NASA Software Engineering Laboratory,  
NASA Goddard Space Flight Center, Greenbelt, MD 20771, USA  
Michael.G.Hinchey@nasa.gov

<sup>3</sup> Advanced Architectures and Automation Branch,  
NASA Goddard Space Flight Center, Greenbelt, MD 20771, USA  
{james.l.rash, Walter.F.Truszkowski}@nasa.gov

<sup>4</sup> Advanced Concepts Business Unit,  
Science Applications International Corp., McLean, VA 22102, USA  
rouffc@saic.com

<sup>5</sup> Virginia Tech, Department of Computer Science,  
Blacksburg, Virginia, USA  
gracanin@vt.edu

**Abstract.** Autonomic Computing (AC), self-management based on high level guidance from humans, is increasingly gaining momentum as the way forward in designing reliable systems to hide complexity and conquer IT management costs. Effectively, AC may be viewed as Policy-Based Self-Management. In this paper we look at the motivation for utilizing NASA requirements-based programming technologies for mechanically transforming policies (expressed in restricted natural language, or appropriate graphical notations) into a provably equivalent formal model that can be used as the basis for code generation and other transformations, with the goal of self-generation of provable autonomic policies.

## 1 Introduction and Motivation

As a rapidly growing field<sup>1</sup>, Autonomic Systems (Autonomic Computing and Autonomic Communications) is a promising new approach for developing large-scale complex distributed computer-based systems. In introducing the concept of Autonomic Computing, IBM's Paul Horn likened the needs of large scale systems management to that of the human Autonomic Nervous System (ANS). The ANS, through the self-regulation, is able to effectively monitor, control and regulate the human body without the need for conscious thought [12]. This self-regulation and separation of concerns provides human beings with the ability to concentrate on high level objectives without having to micro-manage the specific details involved.

---

<sup>1</sup> Consider, e.g., the IEEE Task Force on Autonomous and Autonomic Systems (TFAAS) as of June 2005. See <http://www.computer.org/tab>.

The vision and metaphor of Autonomic Computing is to apply the same principles of self-regulation and complexity-hiding to the design of computer-based systems, in the hope that eventually computer systems can achieve the same level of self-regulation as the human ANS [12][21]. In his talk, Horn highlighted that the Autonomic Computing system must “find and generate rules for how best to interact with neighboring systems” [12]. The majority of current efforts are on the ‘how’ of autonomic systems, such as defining autonomic managers that together with the component that is to be managed make up an autonomic element to exist in a collaborative autonomic environment to provide self-management of the system. Much less is being done on generating the rules and policies that will drive autonomic systems.

The initial long term strategic vision highlighted an overarching self-managing vision where the system would have such a level of ‘self’ capability that a senior (human) manager in an organization could specify business policies, such as profit margin on a specific product range or system quality of service for a band of customers, and the computing systems would do the rest. It has been argued that for this vision to become a reality would require AI completeness, Software Engineering completeness and so on [2]. What is clear in this vision is the importance of policies to empower the system at all levels to self-manage.

## 2 Policy Based Management

Policies have been described as a set of considerations designed to guide decisions of courses of action [17] and policy-based management may be viewed as an administrative approach to systems management that establishes rules in advance to deal with situations that are likely to occur. From this perspective policy-based management works by controlling access to and setting priorities for the use of information and communications technology (ICT) resources<sup>2</sup>, for instance, where a (human) manager may simply specify the business objectives and the system will make it so in terms of the needed ICT [16] for example [13]:

1. “The customer database must be backed up nightly between 1 a.m. and 4 a.m.”,
2. “Platinum customers are to receive no worse than 1-second average response time on all purchase transactions”,
3. “Only management and the HR senior staff can access personnel records”, and
4. “The number of connections requested by the Web application server cannot exceed the number of connections supported by the associated database.”

These examples highlight the wide range and multiple levels of policies available, the first concerned with system protection through backup, the second with system optimization to achieve and maintain a level of quality of service for key customers; while the third and fourth are concerned with system configuration and protection. With one definition of Autonomic Computing being Self-Management based on high level guidance from humans [15] and considering IBM’s high-level set of self-properties (self-CHOP, configuration, healing, optimisation and protection) against the types of typical

<sup>2</sup> See, e.g., Whatis.com, Online computer and internet dictionary and encyclopedia.

policies mentioned previously (optimization, configuration and protection), the importance and relevance of policies for achieving autonomicity becomes clear.

Policy-based management (PBM) has been the subject of extensive research in its own right. The Internet Engineering Task Force (IETF) has investigated policy-based networking as a means for managing IP-based multi-service networks with quality of service guarantees. More recently, PBM has become extremely popular within the telecom industry, for next generation networking, with many vendors announcing plans and introducing products. This is driven by the fact that policy has been recognized as a solution to manage complexity, and to guide the behaviour of a network or distributed system through high-level user-oriented abstractions [18]. A policy-based management tool may also reduce the complexity of product and system management by providing uniform cross-product policy definition and management infrastructure [5].

### 3 Formal Requirements Based Programming

The need for ultra-high dependability systems increases continually, along with a correspondingly increasing need to ensure correctness in system development. By “correctness”, we mean that the implemented system is equivalent to the requirements, and that this equivalence can be proved mathematically. Today there is no automated means of producing a system or a procedure that is a provably correct implementation of the customer’s requirements. Further, requirements engineering as a discipline has yet to produce an automated, mathematics-based process for requirements validation.

Development of a system that will have a high level of reliability requires the developer to represent the system as a formal model that can be proven to be correct. Through the use of currently-available tools, the model can then be automatically transformed into code with minimal or no human intervention. This serves to reduce the risk of inadvertent insertion of errors by developers. Automatically producing the formal model from customer requirements would further reduce the chance of insertion of errors by developers.

Requirements-Based Programming refers to the development of complex software (and other) systems, where each stage of the development is fully traceable back to the requirements given at the outset. Model-Based Development holds that emphasis should be placed on building a model of the system with such high quality that automatic code generation is viable. While this has worked well, and made automatic code generation feasible, there is still the large analysis-specification gap that remains unaddressed. Requirements-Based Programming addresses that issue and ensures that there is a direct mapping from requirements to design, and that this design (model) may then be used as the basis for automatic code generation. In essence, Requirements-Based Programming takes Model-Based Development and adds a front end [9][20].

There have been calls for the community to address Requirements-Based Programming, as it offers perhaps the most promising approach to achieving correct systems [16]<sup>3</sup>. Although the use of Requirements-Based Programming does not specifically pre-

<sup>3</sup> D. Harel. Comments made during presentation at “Formal Approaches to Complex Software Systems” panel session. ISoLA-04 First International Conference on Leveraging Applications of Formal Methods, Paphos, Cyprus. 31 October 2004.

suppose the existence of an underlying formalism, the realization that proof of correctness is not possible without formalism [3] certainly implies that Requirements-Based Programming should be formal. In fact, Formal Requirements-Based Programming, coupled with a graphical representation for system requirements (e.g., UML use cases) possesses the features and advantages of a visual formalism described by Harel [6].

The remainder of this paper describes a method for mechanically transforming system requirements into a provably equivalent formal model that can be used as the basis for code generation and other transformations. The method is applicable to development of policy-based management systems, which, as stated above, is an important part of autonomic systems. In addition, due to the complexity of many policies, development of this part of an autonomic system is crucial to the correct operation of the system and can be very labor intensive. Developing and verifying the policies in an autonomic system in a cost effective manner will be critical for the correct operation of these systems.

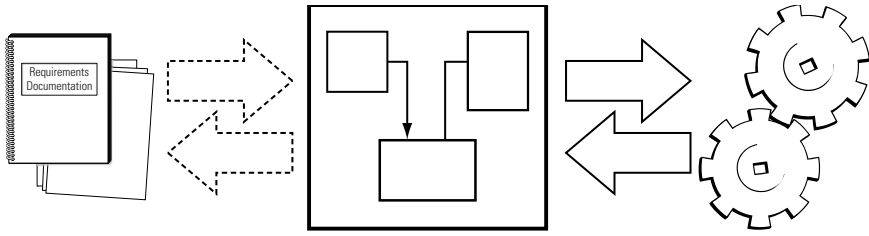
## 4 R2D2C

Our experience at NASA Goddard Space Flight Center (GSFC) has been that while engineers are happy to write descriptions as natural language scenarios, or even using semi-formal notations such as UML use cases, they are loath to undertake formal specification. Absent a formal specification of the system under consideration, there is no possibility of determining any level of confidence in the correctness of an implementation. More importantly, we must ensure that this formal specification fully, completely, and consistently captures the requirements set forth at the outset. Clearly, we cannot expect requirements to be perfect, complete, and consistent from the outset, which is why it is even more important to have a formal specification, which can highlight errors, omissions, and conflicts. The formal specification must also reflect changes and updates from system maintenance as well as changes and compromises in requirements, so that it remains an accurate representation of the system.

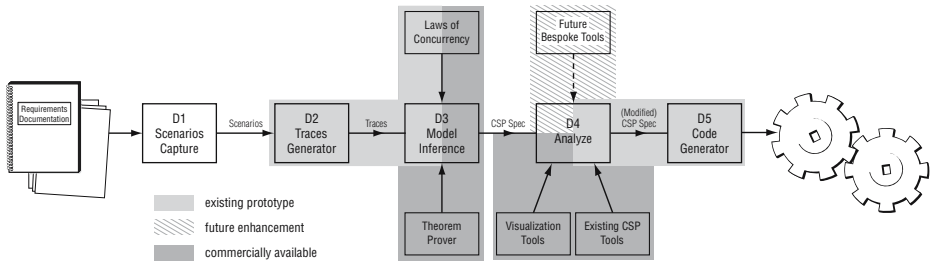
R2D2C, or Requirements-to-Design-to-Code [8][19], is a NASA patent-pending approach to Requirements-Based Programming that provides a mathematically tractable round-trip engineering approach to system development. In R2D2C, engineers (or others) may write specifications as scenarios in constrained (domain-specific) natural language, or in a range of other notations (including UML use cases). These will be used to derive a formal model (Figure 1) that is guaranteed to be equivalent to the requirements stated at the outset, and which will subsequently be used as a basis for code generation. The formal model can be expressed using a variety of formal methods. Currently we are using CSP, Hoare's language of Communicating Sequential Processes [10][11], which is suitable for various types of analysis and investigation, and as the basis for fully formal implementations as well as for use in automated test case generation, etc.

R2D2C is unique in that it allows for full formal development from the outset, and maintains mathematical soundness through all phases of the development process, from requirements through to automatic code generation. The approach may also be used for reverse engineering, that is, in retrieving models and formal specifications from existing code, as shown in Figure 1. The approach can also be used to "paraphrase" (in natural language, etc.) formal descriptions of existing systems.





**Fig. 1.** The R2D2C approach, generating a formal model from requirements and producing code from the formal model, with automatic reverse engineering



**Fig. 2.** The entire process with D1 thru D5 illustrating the development approach

This approach is not limited to generating high-level code. It may also be used to generate business processes and procedures, and we have been experimenting with using it to generate instructions for robotic devices that were to be used on the Hubble Robotic Servicing Mission (HRSM), which, at the time of writing, has not received a final go-ahead. We are also experimenting with using it as a basis for an expert system Verification tool, and as a means of capturing domain knowledge for expert systems, and most recently for generating policies from requirements.

#### 4.1 R2D2C Technical Approach

The R2D2C approach involves a number of phases, which are reflected in the system architecture described in Figure 2. The following describes each of these phases.

**D1 Scenarios Capture:** Engineers, end users, and others write scenarios describing intended policies. The input scenarios may be represented in a constrained natural language using a syntax-directed editor, or may be represented in other textual or graphical forms. Scenarios effectively describe policies that must be adhered to. They describe who various situations and events are to be handled. At the lower (micro) level, these may describe policies of an individual autonomic element. At the overall (macro) level, they may describe policies for a complete system. Policies may be viewed as being analogous to requirements, but are likely to be expressed

at differing levels, and to express a mixture of both functional and non-functional requirements that must be implemented in order to satisfy the policies.

**D2 Traces Generation:** Traces and sequences of atomic events are derived from the scenarios defined in phase D1.

**D3 Model Inference:** A formal model, or formal specification, expressed in CSP is inferred by an automatic theorem prover, in this case, ACL2 [14], using the traces derived in phase D2. A deep<sup>4</sup> embedding of the laws of concurrency [7] in the theorem prover gives it sufficient knowledge of concurrency and of CSP to perform the inference. The embedding will be the topic of a future paper.

**D4 Analysis:** Based on the formal model, various analyses can be performed, using currently available commercial or public domain tools, and specialized tools that are planned for development. Because of the nature of CSP, the model may be analyzed at different levels of abstraction using a variety of possible implementation environments. This will be the subject of a future paper.

**D5 Code Generation:** The techniques of automatic code generation from a suitable model are reasonably well understood. The present modeling approach is suitable for the application of existing code generation techniques, whether using a tool specifically developed for the purpose, or existing tools such as FDR [1], or converting to other notations suitable for code generation (e.g., converting CSP to B [4]) and then using the code generating capabilities of the B Toolkit.

## 4.2 A Simple Example

The Lights-Out Ground Operating System (LOGOS) is a proof-of-concept NASA system for automatic control of ground stations when satellites pass overhead and under their control. The system exhibits both autonomous and autonomic properties [23] [22], and operates by having a community of distributed autonomous software modules work cooperatively based on policies to perform the functions previously undertaken by human operators using traditional software tools, such as orbit generators and command sequence planners. We will not consider the entire LOGOS/ANTS related system here. Although a relatively small system, it is too extensive to illustrate in its entirety in this paper. We will take an example agent, the Pager agent, and illustrate its mapping from natural language descriptions through to the CSP model that can be used to generate code.

Based on defined policies for the operation of the system, the Pager agent sends pages to engineers and controllers when there is a spacecraft anomaly. For example, the Pager agent receives requests from the user interface agent that no analyst is logged on, so it gets paging information from the Database agent and pages an appropriate analyst, and, when instructed by the user interface agent stops paging the analyst. These policies can be stated as follows:

- When the Pager agent receives a request from the User Interface agent, the Pager agent sends a request to the Database agent for an analyst’s pager information and puts the message in a list of requests to the Database agent

---

<sup>4</sup> “Deep” in the sense that the embedding is semantic rather than merely syntactic.

- When the Pager agent receives a pager number from the Database agent, then the Pager agent removes the message from the paging queue and sends a message to the analyst’s pager and adds the analyst to the list of paged people
- When the Pager agent receives a message from the user interface agent to stop paging a particular analyst, the Pager agent sends a stop-paging command to the analyst’s pager and removes the analyst from the paged list
- When the Pager agent receives another kind of message, reply to the sender that the message was not recognized

The above policies for handling anomalies would then be translated into CSP. The following is a partial CSP description of the Pager agent:

```

PAGER_BUSdb_waiting, paged = pager.In?msg →
  case
    GET_USER_INFOdb_waiting, paged, pagee, text
      if msg = (START_PAGING, specialist, text)

    BEGIN_PAGINGdb_waiting, paged, in_reply_to_id(msg), pager_num
      if msg = (RETURN_DATA.pager_num)

    STOP_CONTACTdb_waiting, paged, pagee
      if msg = (STOP_PAGING, pagee)

    pager.Iout!(head(msg), UNRECOGNIZED)
      → PAGER_BUSdb_waiting, paged
  otherwise

```

This specification states that the process *PAGER\_BUS* receives a message on its “*In*” channel and stores it in a variable called “*msg*”. Depending on the contents of the message, one of four different processes is executed based on the policies. If the message is of type *START\_PAGING*, then the *GET\_USER\_INFO* process is called with parameters of the specialist to page (*pagee*) and the text to send. If the message is of type *RETURN\_DATA* with a *pagee*’s pager number, then the database has returned a pager number and the *BEGIN\_PAGING* process is executed with a parameter containing the original message id (used as a key to the *db\_waiting set*) and the passed pager number. The third type of message that the Pager agent might receive is one of type *STOP\_PAGING*. This message contains a request to stop paging a particular specialist (stored in the *pagee* parameter). When this message is received, the *STOP\_PAGING* process is executed with the parameter of the specialist type. If the Pager agent receives any other message than the above three messages, an error message is returned to the sender of the message (which is the first item of the list) stating that the message is “*UNRECOGNIZED*”. After this, the *PAGER\_BUS* process is again executed.

The formal model derived (in CSP) now embodies the policy for anomaly resolution that was specified in the scenarios.

### 4.3 Advantages of the R2D2C Approach

We have not yet had an opportunity to apply R2D2C to policy generation, although that is certainly our plan. In addition to applying it to the HRSM procedures [19], we have applied R2D2C to LOGOS, a NASA prototype Lights-Out Ground Operating System, that exhibits both autonomous and autonomic properties [22][23]. We illustrate the use of a prototype tool to apply R2D2C to LOGOS in [20], and describe our success with the approach.

Here, we summarize some benefits of using R2D2C, and hence of using Formal Requirements-Based Programming in system development. It is our contention that R2D2C, and other approaches that similarly provide mathematical soundness throughout the development lifecycle, will:

- Dramatically increase assurance of system success by ensuring
  - completeness and consistency of requirements
  - that implementations are true to the requirements
  - that automatically coded systems are bug-free; and that
  - that implementation behavior is as expected
- Decrease costs and schedule impacts of ultra-high dependability systems through automated development
- Decrease re-engineering costs and delays

## 5 Conclusions

Autonomic Computing, Self-Management based on high level guidance from humans, has been gaining ground as a significant new paradigm to facilitate the creation of self-managing systems to deal with the ever increasing complexity and costs inherent in today's (and tomorrow's) systems. Policies and policy based management is a key enabling technology for achieving autonomicity. This paper described a method that can produce fully (mathematically) tractable development of policies for autonomic systems from requirements through to code generation. The use of this method was illustrated through an example showing how user formulated policies can be translated into a formal model which can then be converted to code. The requirements-based programming method described will allow faster, higher quality development and maintenance of autonomic systems based on user formulation of policies.

## Acknowledgements

Part of this work has been supported by the NASA Office of Systems and Mission Assurance (OSMA) through its Software Assurance Research Program (SARP) project, Formal Approaches to Swarm Technologies (FAST), and by NASA Goddard Space Flight Center, Software Engineering Laboratory (Code 581). At the University of Ulster by the Centre for Software Process Technologies (CSPT), funded by Invest NI through the Centres of Excellence Programme, under the EU Peace II initiative.

## References

1. *Failures-Divergences Refinement: User Manual and Tutorial*. Formal Systems (Europe), Ltd., 1999.
2. O. Babaoglu, A. Couch, G. Ganger, P. Stone, M. Yousif, and J. Kephart. Panel: Grand challenges of autonomic computing. In *Proceedings of the 2nd IEEE International Conference on Autonomic Computing (ICAC-05)*, Seattle, WA, June 2005.
3. F. L. Bauer. A trend for the next ten years of software engineering. In H. Freeman and P. M. Lewis, editors, *Software Engineering*, pages 1–23. Academic Press, 1980.
4. M. J. Butler. *csp2B : A Practical Approach To Combining CSP and B*. Declarative Systems and Software Engineering Group, Department of Electronics and Computer Science, University of Southampton, February 1999.
5. A. G. Ganek. Autonomic computing: implementing the vision. Keynote presentation, Autonomic Computing Workshop, AMS 2003, 25 June 2003.
6. D. Harel. On visual formalisms. *Communications of the ACM*, 31(5):514–530, May 1988.
7. M. G. Hinchey and S. A. Jarvis. *Concurrent Systems: Formal Development in CSP*. International Series in Software Engineering. McGraw-Hill International, London, UK, 1995.
8. M. G. Hinchey, J. L. Rash, and C. A. Rouff. Requirements to design to code: Towards a fully formal approach to automatic code generation. Technical Report TM-2005-212774, NASA Goddard Space Flight Center, Greenbelt, MD, USA, 2004.
9. M. G. Hinchey, J. L. Rash, W. F. Truszkowski, C. A. Rouff, and R. Sterritt. You can't get there from here! Problems and potential solutions in developing new classes of complex systems. In *Proc. Eighth International Conference on Integrated Design and Process Technology (IDPT)*, Beijing, China, 13–17 June 2005. The Society for Design and Process Science.
10. C. A. R. Hoare. Communicating sequential processes. *Communications of the ACM*, 21(8):666–677, 1978.
11. C. A. R. Hoare. *Communicating Sequential Processes*. Prentice Hall International Series in Computer Science. Prentice Hall International, Englewood Cliffs, NJ, 1985.
12. P. Horn. Autonomic computing: IBM's perspective on the state of information technology. Technical report, IBM T. J. Watson Laboratory, October 15, 2001.
13. D. Kaminsky. An introduction to policy for autonomic computing. white paper, March 2005.
14. M. Kaufmann and Panagiotis Manolios and J Strother Moore. *Computer-Aided Reasoning: An Approach*. Advances in Formal Methods Series. Kluwer Academic Publishers, Boston, 2000.
15. J. O. Kephart and W. E. Walsh. An artificial intelligence perspective on autonomic computing policies. In *Proc. Fifth IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'04)*, pages 3–12, 7–9 June 2004.
16. L. Lymberopoulos, E. Lupu, and M. Sloman. An adaptive policy-based framework for network services management. *Journal of Network and Systems Management*, 11(3), 2003.
17. M. J. Masullo and S. B. Calo. Policy management: An architecture and approach. In *Proc. IEEE First International Workshop on Systems Management*, Los Angeles, California, USA, 14–16 April 1993.
18. A. Meissner, S. B. Musunoori, and L. Wolf. MGMS/GML—towards a new policy specification framework for multicast group integrity. In *Proc. 2004 International Symposium on Applications and the Internet (SAINT2004)*, Tokyo, Japan, 2004.
19. J. L. Rash, M. G. Hinchey, C. A. Rouff, and D. Gračanin. Formal requirements-based programming for complex systems. In *Proc. International Conference on Engineering of Complex Computer Systems*, Shanghai, China, 16–20 June 2005. IEEE Computer Society Press, Los Alamitos, Calif.

20. J. L. Rash, M. G. Hinchey, C. A. Rouff, D. Gračanin, and J. D. Erickson. A tool for requirements-based programming. In *Proc. International Conference on Integrated Design and Process Technology (IDPT 2005)*, Beijing, China, 13–17 June 2005. The Society for Design and Process Science.
21. R. Sterritt. Towards autonomic computing: Effective event management. In *Proc. 27th Annual IEEE/NASA Software Engineering Workshop (SEW)*, pages 40–47, Greenbelt, Maryland, USA, 3–5 December 2002. IEEE Computer Society Press, Los Alamitos, Calif.
22. W. F. Truszkowski, M. G. Hinchey, J. L. Rash, and C. A. Rouff. Autonomous and autonomic systems: A paradigm for future space exploration missions. *IEEE Transactions on Systems, Man and Cybernetics, Part C*, 2006 (to appear).
23. W. F. Truszkowski, J. L. Rash, C. A. Rouff, and M. G. Hinchey. Some autonomic properties of two legacy multi-agent systems — LOGOS and ACT. In *Proc. 11th IEEE International Conference on Engineering Computer-Based Systems (ECBS), Workshop on Engineering Autonomic Systems (EASe)*, pages 490–498, Brno, Czech Republic, May 2004. IEEE Computer Society Press, Los Alamitos, Calif.

# Intrusion Detection with CUSUM for TCP-Based DDoS

Fang-Yie Leu and Wei-Jie Yang

Department of Computer Science and Information Engineering,  
Tunghai University, Taiwan  
leufy@thu.edu.tw

**Abstract.** DDoS(Distributed Denial of Service) is the most troublesome attack nowadays, especially for those people whose operational environment relies on network services and/or the Internet. However, attackers often penetrate innocent routers and hosts to make them unwittingly participate in such a large scale attack as zombies or reflectors. In this paper, we propose an Intrusion Detection System (IDS), named CUSUM Intrusion Detection System (CIDS), which invokes CUSUM as its detection algorithm and logically divides Internet into many autonomous network management units (NMUs), each deploys a CIDS to discover attacks and identify what role a client in such an attack acts as.

## 1 Introduction

As Internet grows quickly, its network security has recently attracted researchers' attention. The threats caused by intruders may be losing disclosing secrets or reducing market opportunity. Intrusion Detection Systems (IDSs) or firewalls are prosperously deployed to protect network systems. Generally, IDS based on detective features can be classified into Network-based, host-based and hybrid of them. However, most of them are behavior-based and most of traditional Network-based IDSs (NIDSs) detect abnormal network behavior by monitoring if network traffic exceeds its threshold or not. However, such systems are no longer appropriate and sufficient nowadays due to the divergence of Internet activities.

DoS (Denial of Service) and DDoS (Distributed Denial of Service) are notorious attacks owing to easy commencement and tremendous destruction. Some popular web sites, such as Yahoo, Amazon, and eBay, had ever been attacked by them in February 2000. These attacks can be simply issued by attacking tools which with friendly user interfaces are available on Internet. Therefore, attackers can easily produce huge and legitimate traffic of the same or different protocols to flood victims. Gibson [1] foretold that at 2:00 AM, January 11th 2002, grc.com would be blasted off by an advanced malicious packet flood. What surprising us was that this attack came from more than 200 non-spoofed core routers. This attack, called Distributed Reflective Denial of Service (DRDoS), as shown in [1] is an extension of the DDoS by deploying some clients as reflectors to launch attack packets. Moreover, SYN\_ACK flooding packets responded from reflectors through HTTP (web) port 80 or some other frequently used ports are hard to distinguish from those of normal connections.

Also, most DDoS and DRDoS attacks are TCP based since TCP Three-way handshake lacks a verification mechanism. In this paper, we proposed a detecting

system, named CUSUM Intrusion Detection System (CIDS), which invokes the Cumulative SUM (CUSUM) algorithm [2, 3] to discover attacks and the role a node acts as within an attack. To develop CIDS, we first analyze TCP-based attacks and the drawback of TCP protocol. Next, abnormal network behaviors occurred at victim, reflector and attacker (zombie) sites during an attack are addressed. Finally the ways to detect DDoS and DRDoS attacks by invoking CUSUM algorithm are designed.

## **2 Related Work**

### **2.1 DDoS Attack**

DoS/DDoS after initiated will continue until it is terminated by hackers or mitigated by victims. Situation becomes worse when their control messages are encrypted to evade IDS's detection. Many countermeasures are proposed to defense against DDoS attack [4, 5]. Some prevent clients from being zombie agents. Some try to relieve victims from attack damage. IP traceback mechanism [6-10] is also one of the major security aspects. Most solutions mainly focus on how to identify traffic sources. The main concerns of designing a detector are less computation and fast detection.

### **2.2 TCP Three-Way Handshake and SYN Flood Attack**

Establishing a TCP session requires the TCP Three-Way Handshake, whereas normally disconnecting a TCP connection needs to exchange four packets [11]. After accepting a SYN request, server allocates resources for the request and replies SYN\_ACK. This is so called half-open connection. A traditional SYN Flood delivers large amount of SYN packets with randomly spoofed source IPs to consume server's resources or network bandwidth. Unfortunately, only few of the spoofed addresses really alive and then send RST to terminate the non-existing connections. Most source addresses reply no SYN-ACK packets resulting in server reserving too many resources so that it is unable to provide services to other legal users.

### **2.3 Distributed Reflective Denial of Service (DRDOS) Attacks**

Paxson [12] had deeply analyzed reflector attacks in early 2001. Any IP host that returns a reply as receiving a packet may act as the reflector. Using traceback techniques, we can trace the reflectors, but can not locate and identify who issues the attack.

## **3 CUSUM Intrusion Detection System (CIDS)**

We first divide Internet into many autonomous Network Management Units (NMUs). An enterprise intranet and a campus network are examples. All ingress and egress packets of an NMU are detected by CIDS, an IDS integrating HIDS(Host-based IDS) and NIDS (Network-based IDS) properties to detect DDoS and DRDoS, as shown in Fig. 1, by means of collecting packets flowing through internal router as the observed event sequence. Like most detection systems, CIDS compares the observed sequence with users' normal behaviors recorded in profiles to find out the significant discrepancy and difference. A procedure, named Change-Point Detection designed to



detect the change point of a network behavior, is as follows. First, compare observed event sequence with user profiles. If any difference is significant, identify the time point the change happens so as to real time discover when the attack starts. Second, CUSUM [2, 13] is deployed to sequentially monitor input random variables. A simple parametric approach [13, 14] is often too simple to accurately model a network session due to the complexity of Internet. CUSUM with the characteristics of sequential and nonparametric light computation load can make CIDS work online.

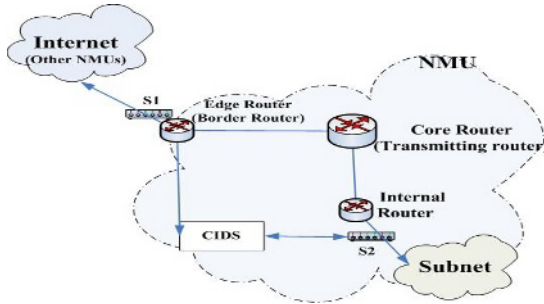


Fig. 1. Framework of a NMU (S1 and S2 are switches)

### 3.1 CUSUM

CUSUM can detect sharp but continuous increase. Some of its assumptions are as follows. First, let  $X_n$  be the packets collected by CIDS in a sampling time  $\Delta n$  and  $\bar{X}$  the mean value of random sequence  $X$ ,  $X = \{X_n, n = 0, 1, 2, \dots\}$ . Second, let  $Z = \{Z_n, n = 0, 1, 2, \dots\}$  with  $a$ , where  $Z_n = X_n - a$  and  $a$  is the peak value of normal traffic for a specific network status so that all elements of  $Z$  are negative, so is  $\bar{Z}$ . When a change, such as flow-based attack, occurs,  $Z_n$  will suddenly increase to positive, as illustrated in Fig. 2.  $Z_k \geq \bar{Z} + h$ , for some  $k$ , indicates an attack possibly starts where  $k$  is the smallest  $n$  and  $h$  the threshold of abnormal network traffic.  $\Delta_k$  is then considered as the change point of network traffic.  $y_{n-1} + Z_n \leq 0$  shows there is no attack. CUSUM accumulates  $Z_n$ ,  $m \geq k$ , with formula(1) which is the recursive version of the non-parametric CUSUM algorithm[2].

$$y_n = (y_{n-1} + Z_n)^+, y_0 = 0 \tag{1}$$

where  $x^+ = x$  if  $x > 0$  and 0 otherwise.  $Z_n, n > k$ , may now be positive or negative.

The decision function at  $\Delta p$ , say  $d_p(y_p)$ , is as follows:

$$d_p(y_p) = \begin{cases} 1 & \text{if } y_p > N; \\ 0 & \text{else} \end{cases} \tag{2}$$

where  $N$  is the threshold of an attack. ‘1’ indicates an attack occurs, while ‘0’ shows network operates normally.

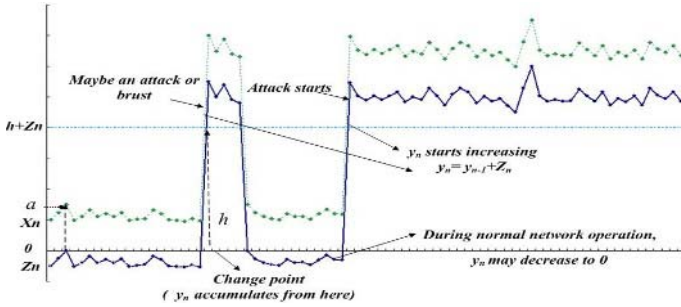


Fig. 2. An example of CUSUM

### 3.2 TCP SYN Flood

In the following, we traditional and reflective TCP Flood attacks in detail. A client inside an NMU may act as a victim, reflector, or an attacker (zombie).

During an attack, a zombie sends a large amount of SYN packets of random source IPs to victim as shown in Fig. 3(a). Due to SYN-ACK packets never replying back,  $|O-SYN|$  and  $|I-SYN\_ACK|$  at zombie significantly differ where  $|X|$  represents the number of  $X$ ,  $O-SYN$  and  $I-SYN\_ACK$  stand for outgoing SYN and incoming SYN-ACK respectively. However,  $|I-SYN|$  and  $|O-SYN\_ACK|$  at victim are similar and huge.

Wang et al. [11] mentioned, before a normal TCP session ended, a SYN packet would result in a returned FIN packet, so does SYN-ACK. Generally, a RST packet following a TCP packet, e.g., SYN, SYN-ACK, URG or PSH, represents one of the three cases: (1) terminating a TCP session; (2) aborting a TCP request; (3) destinating a packet to a closed port or an unconnected node.

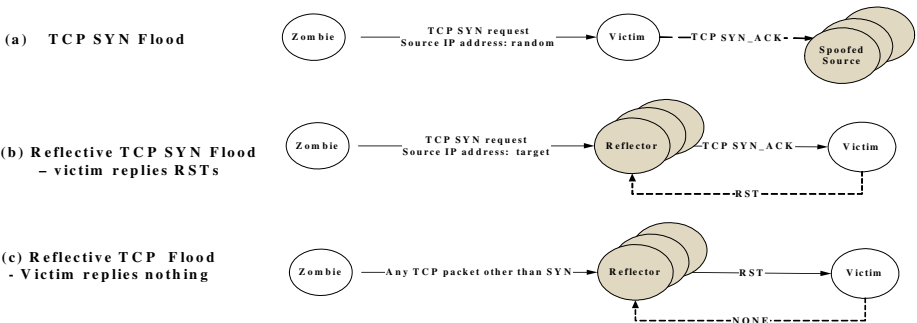


Fig. 3. Attack nodes of TCP Flooding

#### (1) Traditional SYN Flood

Normally,  $|SYN|+|SYN/ACK|$  almost equals to  $|RST|+|FIN|$  at a node. The ratio of active RST packets, generated by the first two cases which are strongly related to

SYN, is about 75% of all RST packets, thus taking 3/4 as our experienced cumulated normal amount. The remaining 1/4 resulted from the third is treated as background noise to improve detection sensitivity and decrease false alarms [11]. CIDS considers SYN-FIN, SYN\_ACK-FIN, and SYN-RST<sub>active</sub> pairs normal behaviors at victim. Those deviated from normal indicate the node is now under a TCP SYN Flood attack.

**(2) Reflective TCP Flood**

During a reflective attack, zombies send some level of volume of TCP SYN requests to each reflector [1]. Two facts are useful in detecting such an attack. First, as receiving a spoofed request, a reflector replies a SYN\_ACK to victim which, as shown in Fig. 3(b), will send a RST back. Second, when any TCP packet other than SYN, such as URG, PSH, FIN or a TCP packet without a flag, arrives without any previous handshake, reflector replies a RST to victim which answers nothing, as illustrated in Fig. 3(c). An abnormal increase of  $|O-RST|$  at victim implies the existence of reflector which may be in underlying or other NMU.

Wang et al. [3, 11] proposed the difference between  $|I-SYN\_ACK|$  and  $|O-SYN|$  is helpful in discovering the existence of attacker in situation of both DDoS and DRDoS. From the second facts stated above, we discover that a  $|I-SYN\_ACK|$  and  $|O-RST|$  packets at a victim hugely increase, so do  $|O-SYN\_ACK|$  and  $|I-RST|$  at a reflector. But  $|O-SYN|$  and traffic generated by other TCP packets at both nodes do not. The phenomenon is extremely useful in detecting reflector and victim.

**3.3 Detecting TCP SYN Flood**

CIDS monitors network behavior in order to collect corresponding  $X_n$ .

**(1) Victim**

Let  $|S_{SYN}|$ ,  $|S_{FIN}|$ ,  $|S_{SYN\_ACK}|$  and  $|S_{RST}|$  respectively represent numbers of SYN, FIN, SYN\_ACK and RST packets observed within a sampling interval  $\Delta n$ , but ignoring packet direction. Let  $X_{Vn}$  be the normalized  $X_n$ , obtained by formula (3), so that  $X_{Vn}$  is independent of network size and ordinary traffic since normally the ratio between request pairs(SYN-SYN\_ACK) and reply pairs(RST-FIN) is relatively stable.

$$X_{Vn} = \frac{(|S_{SYN}| + |S_{SYN\_ACK}|) - (|S_{FIN}| + |S_{RST}|)}{(|S_{FIN}| + |S_{RST}|)} \tag{3}$$

With  $X_{Vn}$ , a victim can be discovered. However, [11] mentioned that most TCP connections lasted 12-19 seconds. In this paper,  $\Delta n=10$ sec, therefore, FIN and RST packets and their corresponding SYN and SYN\_ACK packets always appear in different intervals. Let the sampling time correction (delay) of  $|S_{FIN}|$  and  $|S_{RST}|$  be  $\beta$ , i.e.,  $\Delta n_{(FIN,RST)} = \Delta n_{(SYN,SYN\_ACK)} + \beta$ , where  $\Delta n_{(FIN,RST)}$  and  $\Delta n_{(SYN,SYN\_ACK)}$  are the sampling time of  $S_{FIN}$  ( $S_{RST}$ ) and  $S_{SYN}$  ( $S_{SYN\_ACK}$ ) respectively. According to [11],  $\beta=10$  can balance the sensitivity and detection accuracy.

Generally, few RST packets are generated under normal operation. The major reasons that  $(|S_{SYN}| + |S_{SYN\_ACK}|)$  and  $(|S_{FIN}| + |S_{RST}|)$  significantly defer are as follows.

- (1) The long-lived TCP connection may result in incomplete SYN-FIN pairs.
- (2) The retransmission of TCP requests (SYN) may also conduct the difference due to connecting to a non-existing or failed node.
- (3) An RST is generated long after its corresponding SYN\_ACK.

Therefore, by monitoring  $X_{Vn}$ , when the parameters  $h$ ,  $a$  and  $N$  of the given  $X_n$ , say  $h_v$ ,  $a_{Vn}$  and  $N_v$ , are defined where  $Z_{Vn} = X_{Vn} - a_{Vn}$ ,  $h_v = 2$  and  $a_{Vn} = 1$  [3], decision function can then determine if a node  $V$  is a victim of a traditional TCP SYN Flood attack or not. Let

$$X_{Vn}' = \frac{|I - SYN\_ACK| - |O - SYN|}{|O - SYN|} \quad (4)$$

Similar to  $X_{Vn}$ , given  $a_{Vn}'$ ,  $h_v'$  and  $N_v'$ , CIDT can decide a node  $V$  is now a victim of a reflective TCP SYN Flood or not. This can be very accurate since intruder can not decrease the value of  $X_{Vn}'$  by generating spoofed O-SYN for  $V$ .

For detecting victim in Fig. 3(c), let

$$X_{Vn}'' = \frac{|I - RST| - avg |I - RST|}{avg |I - RST|} \quad (5)$$

where  $avg$  represents average.

### (2) Attacker (Zombie)

At the attacker side, let

$$X_{Zn} = \frac{|O - SYN| - |I - SYN\_ACK|}{|I - SYN\_ACK|} \quad (6)$$

Given  $h_{Zn}$ ,  $a_{Zn}$  and  $N_{Zn}$ , CUSUM can determine if a node is zombie or not for both of traditional and reflective TCP SYN Flood attacks.

### (3) Reflector

Let

$$X_{Rn} = \frac{|I - RST| - avg |I - RST|}{avg |I - RST|} \quad (7)$$

Given  $h_R$ ,  $a_R$  and  $N_R$ , a reflector of a reflective TCP SYN Flood can be detected. However, observing  $X_{Rn}$  may be insufficient since its  $|I - RST|$  may be not huge enough, especially when many reflectors are deployed. An abnormal increase of both  $X_{Rn}$  and  $X_{Zn}$ , in the same or different NMUs, indicate that reflectors exist. This finding can be the basis for CIDS to discover who the reflectors are. However, as reflector and zombie are located in different NMUs, they have to communicate with each other [15].

Besides, to detect reflector with situation in Fig. 3 (c), let

$$X_{Rn}' = \frac{|O - RST| - \text{avg} |O - RST|}{\text{avg} |O - RST|} \tag{8}$$

Given  $h_{Rn}'$ ,  $a_{Rn}'$ , and  $N_{Rn}'$ , reflector can be then detected. Combining  $X_{Rn}$  with  $X_{Rn}'$ , CIDS can detect a reflector by monitoring  $|S_{RST}|$ .

Besides, supporting factor to discriminate the reflector of Fig. 3(b) and victim of Fig. 3(c) is required. Let

$$X_D = \frac{|S_{RST}| + |S_{FIN}|}{|S_{SYN}| + |S_{SYN}|} \tag{9}$$

$X_D$  is a stable under normal network operation. During a reflective TCP SYN Flood as the one in Fig. 3(b),  $|S_{SYN}|$ ,  $|S_{SYN\_ACK}|$  and  $|S_{RST}|$  increase, but  $|S_{FIN}|$  does not, and  $|I - RST|$  is almost equal to  $|O - SYN\_ACK|$ . Thus  $X_D$  does not increase. But in Fig. 3(c), only  $|I - RST|$  becomes large resulting in abnormal increase of  $X_D$ .

### 4 Defending with Autonomous NMU

CIDS is originally deployed as the detection component of Intrusion Forecast and Traceback System (IFTS) [15] which integrates intrusion detection and traceback and provides some policies against DDoS attack. An NMU deploys an IFTS as its security system of which a hash-based Traceback mechanism[6] is developed so that tracing intruders can be performed once one attack packet has been detected.

Besides, an IFTS has an Intrusion Response Manager (IRM), the communication center of an NMU, offering the Certification Authority for exchanging information among NMUs. The information consists of tracing messages and a request for filtering attack packets. Thus, NMUs can defeat DDoS through collaborative cooperation with others, especially by deploying CIDS as its powerful DDoS/ DRDoS detector.

Detecting malicious behavior, CIDS analyzes abnormal immediately. Real attackers can be found through the MAC Address rather than spoofed IP addresses since we gather packet information before it flows through the first router of an NMU [15].

Once a suspected attacker, reflector or a victim is discovered, CIDS asks IRM to notify the victim's NMU, may be local or remote, which, after confirming some packets coming from the attacker, in turn asks its traceback mechanism to trace to the real zombies and to filter out attack packets in order to mitigate damage.

### 5 Experiments

Our experimental environment is as follows. The observed interval  $\Delta n_{(FIN,RST)} = \Delta n_{(FIN,RST)} = \beta = 10$  sec. Also, normally the  $|O - SYN|$  at a node is always larger than  $|I - SYN\_ACK|$  due to the SYN loss and subsequent retransmission of a

TCP SYN request. Although Wang et al. [3, 11] treated them as white noise. But they cause negative  $\overline{X_{Vn}}$  under normal network operation, as shown in Fig. 4. Therefore, let  $Z_{Vn} = X_{Vn}$ , i.e.,  $a=0$ .

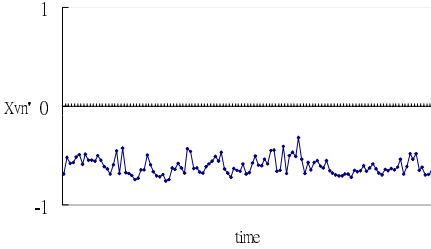


Fig. 4.  $X_{Vn}$  with negative  $\overline{X_{Vn}}$  (Reflective TCP SYN Flood)

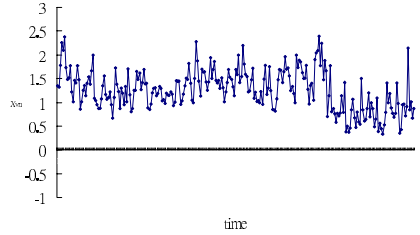


Fig. 5. Example of  $X_{Vn}$  (TCP SYN Flood)

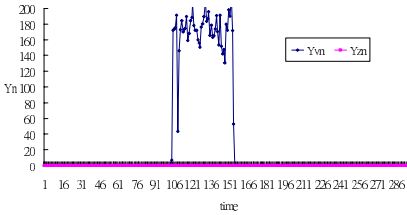


Fig. 6.  $y_{Vn}$  under DDoS attack

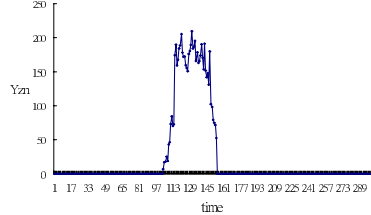


Fig. 7.  $y_{Zn}$  under DDoS

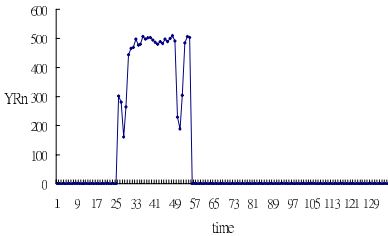


Fig. 8.  $y_{An}$  during an DRDoS

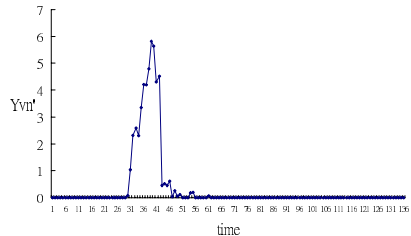


Fig. 9.  $y_{Vn}$  under DRDoS

Due to loss of SYN packets, we give the customized  $h_z = 4$ ,  $a_{Zn} = 2.5$  and  $N_z = 4$  for  $X_{Vn}$  according to our offline observation shown in Fig. 5. The system with the properties stated above is involved as our testbed.

The first experiment is a DDoS attack. Victim and zombie are located in different NMUs, say  $NMU_v$  and  $NMU_z$  respectively. We can see that  $y_{vnt}$  shown in Fig. 6

increases to huge immediately, but  $y_{Zn}$  of each node in  $NMU_V$  does not, showing that the attack comes from other NMU. This occurrence can identify a victim from local nodes. Within  $NMU_Z$ , some node's  $y_{Zn}$  increases to exceed  $N_z$ , as illustrated in Fig. 7, indicating this node is an attacker.

The second experiment is a DRDoS attack. Some  $X_{Rn}$  ' in  $NMU_R$  increases and  $y_{Rn}$  soon exceeds  $N_R$ , as shown in Fig. 8, indicating it is a reflector. Fig. 9 illustrates that  $y_{Vn}$  'in  $NMU_V$  also exceeds its threshold as an attack happens, stating that victim is now under attack.

## 6 Conclusion

With customized CUSUM parameters, all the situations in NMU can be totally monitored by CIDS with the decision function  $d_n(y_n)$  to quickly detect the role a host may act as in real time. The attack can be defeated in the initial stage since the zombie can be located by its local CIDS.

Besides, detecting reflector can help original IFTS traceback mechanism to overcome tracing limitation. Attacking sources in a DRDoS attack can be then traced after reflectors are identified, whereas logical attacks can be traced directly.

With the assist of lightweight detecting approach, like CUSUM, and the help of traceback and filtering mechanisms, we can mitigate the victim from the progressing attack. CIDS can offer us more defending power against the distributed malicious behavior so as to carry out a more secure Internet.

## References

1. Gibson, S.: DRDoS, Distributed Reflection Denial of Service. <http://grc.com/dos/drdo.htm>.
2. Brodsky, B.E., Darkhovsky, B.S.: *Nonparametric Methods in Change-point Problems*. Kluwer Academic Publishers (1993).
3. Wang, H., Zhang D., Shin, K.G.: Change-Point Monitoring for the Detection of DoS Attacks. *IEEE Transactions on Dependable and Secure Computing*, vol. 1, (Oct.-Dec. 2004) 193- 208.
4. Chang, R.K.C.: Defending against flooding-based distributed denial-of-service attacks: a tutorial. *IEEE Communications Magazine*, vol. 40 ( Oct. 2002) 42-51.
5. Leu, F.Y., Hung J.J., Hung, C.H.: UDAIDTS : Union Defense with Active Intrusion Detection and Hash-based Traceback System. *Proceedings of International Conference on Information Management Conference*, Taiwan (July 2003) (in Chinese).
6. Snoeren, A.C. et al.: Single-Packet IP Traceback. *IEEE/ACM Transaction on Network*, vol. 10, no. 6 (Dec. 2002) 721-734.
7. Belenky, A., Ansar, N.: On IP traceback. *IEEE Communications Magazine*, vol. 41 (July 2003) 142-153.
8. Dawn Song, X.D., Perrig, A.: Advanced and Authenticated Marking Schemes for IP Traceback. *Proceedings of 20th Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 2 (April 2001) 878-886.

9. Bellovin, S.M.: ICMP Traceback Messages. IETF draft, 2000. <http://www.research.att.com/smb/papers/draftbellovin-itrace-00.txt>, (refer to in March 2005).
10. Savage, S. et al.: Network Support for IP Traceback. *IEEE/ACM Transaction on Network*, vol. 9, no. 3 (June 2001) 226-237.
11. Wang, H., Zhang, D., Shin, K.G.: Detecting SYN Flooding Attacks. *Proceedings of IEEE Computer Communications* (June 2002) 1530-1539.
12. Paxson, V.: An Analysis of Using Reflectors for Distributed Denial-of Service Attacks. *Computer Communication Review*, 31(3) (July 2001) 38-47.
13. Mirkovic, J., Prier, G., Reiher, P.: Attacking DDoS at the Source. *Proceedings of IEEE International Conference on Network Protocols* (Nov. 2002) 312-321.
14. Gil, T.M., Poletter, M.: MULTOPS: A Data-Structure for Bandwidth Attack Detection. *Proceedings of USENIX Security Symposium* (Aug. 2001).
15. Leu, F.Y., Yang, W.J., Chang, W.K.: IFTS: Intrusion Forecast and Traceback based on Union Defense Environment. *Proceedings of the 11th International Conference on Parallel and Distributed Systems* (July 2005).



# A Digital Content Distribution Using a Group-Key and Multi-layered Structure Based on Web

Yun-Ji Na<sup>1</sup> and Il Seok Ko<sup>2</sup>

<sup>1</sup> Department of Internet Software, Honam University 59-1, Seobong-Dong,  
Gwangsan-Gu, Gwangju 506-741, South Korea  
yjna@honam.ac.kr

<sup>2</sup> School of Computer and Industrial Engineering, Yonsei University 134,  
Shinchon-Dong, Seodaemun-Gu, Seoul 120-749, South Korea  
isko@ctech.ac.kr

**Abstract.** Regarding the design of a multimedia digital content distribution system, the important issues are to supply a large amount of multimedia digital content to users and to guarantee the security of digital content. In this study we proposed designing a security technique for each group in a multilayered structure, and on a caching technique, which is based on this security technique, and to improve the user's response speed. Using these techniques guarantees the security of digital content distribution.

**Keywords:** web based system, multimedia digital content distribution, multilayered structure.

## 1 Introduction

Web based services have been activated due to the increase in network speed. Also, there is no exception in the field of digital content, in which distribution of digital content has rapidly increased [1,2]. However, almost all Web services throughout the Web have security problems due to their specific media characteristics in the Web itself. Due to this security problem, studies on security techniques have been increasingly stressed. Studies on the security techniques based on the Web consists of implementing a type of basic security technique itself [3,4], and of application techniques for the application of Web services [2,5,6].

Recent studies on the transmission of digital content have been focused on the guarantee of safety and effective distribution. However, the improvement of transmission delay is also considered with this safety guarantee in the transmission of multimedia digital content. Thus, the major issue in the design of a multimedia digital content through the Web can be defined as a guarantee of the security of digital content, and fast supplement of a large amount of multimedia digital content to the user.

A content acceleration technique used in the Web is a type of user response time (web browser response time) and network traffic saving technique. In order to perform this content acceleration, a web caching method is used [7,8,9]. A web caching method increases the efficiency of fast response and network use by saving web objects, which are required by the user, who is geographically located at a close position to the Internet. Studies on the CDN (Content Delivery Network) have been increasingly

stressed to effectively distribute digital content in the Web, in which an application of the caching technique can increase the system efficiency in a system design process. It is necessary to design a system, which reflects the characteristics of multimedia digital content, in order to increase the performance of content acceleration using a caching technique in the transmission of multimedia digital content.

In this study we proposed designing a security technique for each group in a multilayered structure, and on a caching technique, which is based on this security technique, and to improve the user's response speed. Using these techniques guarantees the security of digital content distribution.

## 2 System Design

### 2.1 System Structure

Fig. 1 presents a conceptual configuration of the system. The DCP (Digital Content Provider) is a supplier of DC (Digital Content). The DCUG (Digital Content User Group) is a user group, which is supplied by DC. Almost all users of multimedia are only interested in a certain passive action. However, a delicate encryption algorithm and certification requires a certain complicated process. This process is the cause of time delay. Thus, it is necessary to consider the transmission of DC from the view points of safety and execution speed.

Because the user of a DCUG, which is a user group of DC, applied in the proposed system can be certified in the DCUG, the user certification becomes fast and easy. In addition, an effect of the Internet traffic of DC in the proposed system decreases, and the execution speed increases due to the fact that the system will be directly affected by the DCUG cache.

Fig. 2 shows the configuration of a DCUG. A DCUG is managed by grouping it in two different groups. The first group is an authorized user group, which has the authority to use encrypted DC, and the second group is a user group, which has no authority to use encrypted DC. In addition, a DCUG uses a digital content accelerator to increase the user response speed.

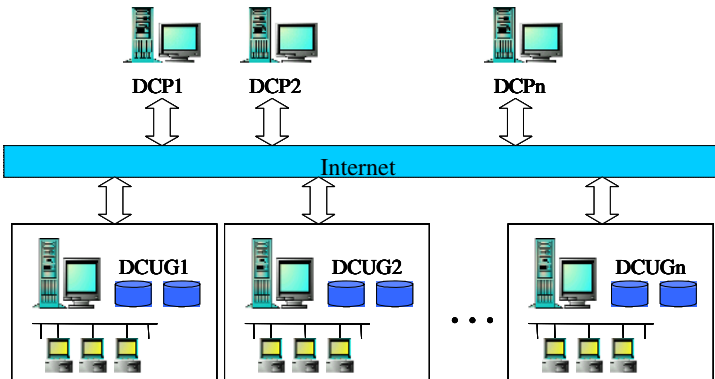


Fig. 1. System Structure

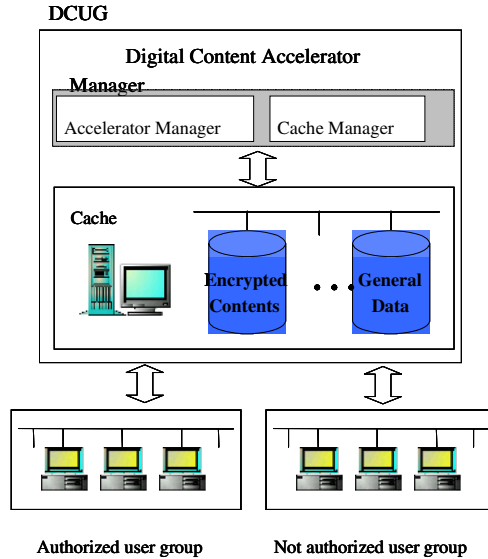


Fig. 2. DCUG Structure

A cache is managed by classifying a caching scope as an authorized user and an unauthorized user. Therefore, the structural security can be managed in the level of system by separating the DC as an authorized DC and an unauthorized DC. In the caching scope of an authorized user, the caching scope can be managed by classifying the DC as an encrypted DC and a generalized DC.

## 2.2 Certification

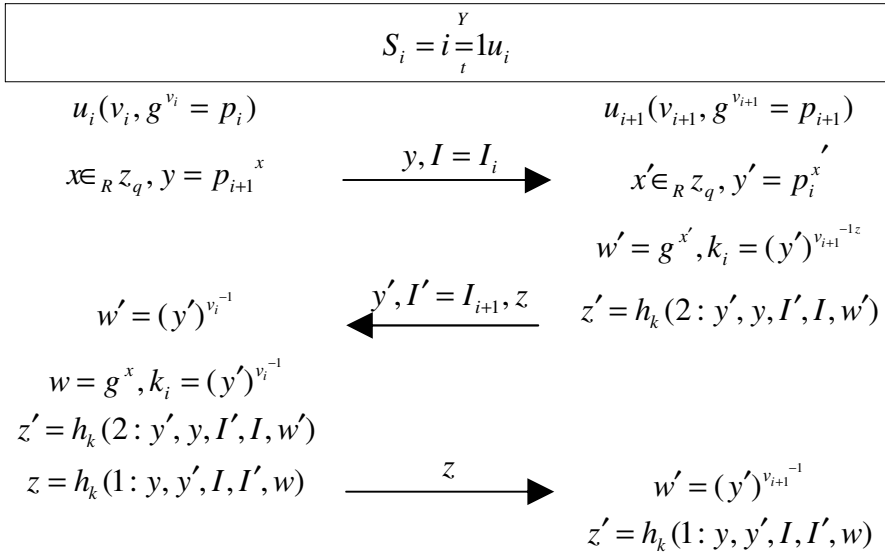
When the authorized user in the DCUG is unable to find the required content in the cache list, the DCUG should receive the content from the appropriate DCP server. In this case, the DCUG and DCP servers should issue a certificate by connecting the CA (Certificate Authority) before transmitting and receiving encrypted data for each other. The issuing process of the certificate is as follows.

- ① Connecting to the CA server,
- ② Requesting a certificate for the CA server,
- ③ The CA server transmits a certificate requirement to the DCUG and DCP servers,
- ④ The DCP and DCUG servers produce a key pair of themselves,
- ⑤ Writing a certificate requirement,
- ⑥ The DCUG and DCP servers transmit their public keys and certificate requirements to the CA server,
- ⑦ The CA server issues a certificate including a public key by verifying the received certificate requirement,
- ⑧ The CA server saves the information of the certificate requirement of the DCUG and DCP servers and certificates to DB,
- ⑨ The CA server transmits certificates of the DCUG and DCP servers to the DCUG and DCP servers,
- ⑩ The DCUG and DCP servers save certificates received from the CA server including their private keys.

In the case of the use of the same key for all the members of the DCUG, this will cause a weakness in the security. Thus, a key agreement between the members of the

DCUG is required. The members of the DCUG calculate the key by themselves. Table 1 presents a key agreement process between the members of the DCUG.

**Table 1.** Key agreement process of the members of in the DCUG



Where, the symbols noted in Table 1 are as follows.

- $u$  : members of the group communication consisted of the DCUG
- $k_i$  : pre-shared key through the key agreement process
- $w_i$  : public information calculated by the input value using a single direction function for the pre-key
- $K_i$  : shared DCUG keys between the members for each DCUG
- $g^{xk_i}$  : shared keys produced by the calculation process for each member and group manager
- $t$  : number of members of the DCUG

In addition, the initial configuration is as follows.

- ①  $p$  : 1024 bits prime number
- ②  $q$  : 160 bits of a prime factor of  $p - 1$
- ③  $g$  : is an element of  $Z_p^*$

The calculation of the modular exponent for the generator of  $g$  is performed in the modulo  $p$ , in which  $h$  is a hash function, and satisfies  $h : \{0,1\}^* \rightarrow \{0,1\}^q$ . In

addition, the member of  $u_i, u_{i+1}$  is a key agreement process, and configures a certificate and pre-shared key. The member of  $u_i$  generates a public key of  $p_i$  using a private key of  $v_i$ , and the member of  $u_{i+1}$  also generates a public key of  $p_{i+1}$  using a private key of  $v_{i+1}$ . Each member calculates  $y, y'$  using their opponent's public keys. This can be used to calculate a pre-shared key of  $k_i = \alpha^{xy}$ .

Finally, the confirmation for the pre-shared key can be performed by transmitting the value of a single direction function, which is produced by a pre-shared key. Thus, the members not only share the pre-shared key safely, but also form reliance between members.

Each member calculates the public information of  $w_i$  using the two shared keys of  $k_i, k_{i-1}$  of the DCUG, in which each member calculates their keys using this public information. The calculation process can be noted as follows.

① The member of  $u_{i-1}, u_i$  generates a pre-shared key using a key agreement process.

② The member of  $u_i$  calculates the public information of  $w_i = h(k_i) - h(k_{i-1})$  using a pre-shared key.

$h(k_i)$  : This applies  $n$  times of single direction functions using the input value of a pre-shared key.

③ The member of  $u_i$  produces a small group key of  $K$  for the members of a small group, which is authorized by applying an inductive method as follows, using the public information.

$u_i$  has the element of  $k_{i-1}, k_i$ , and the small group key of is configured by the equation as follows.

$$K = h(k_i) + h(k_{i-1}) + \dots + h(k_1) \quad (\text{where, } 1 \leq l \leq n)$$

Because  $u_i$  recognizes the value of  $h(k_{i-1}), h(k_i)$ , the public information of  $u_{i+1}$  can be calculated using the equation of  $w_{i+1} = h(k_i) - h(k_{i+1})$ .

Because  $u_i$  recognizes the value of  $h(k_{i-1}), h(k_i)$ , the public information of  $u_{i-1}$  can be calculated using the equation of  $w_{i-1} = h(k_{i-2}) - h(k_{i-1})$ .

### 2.3 Transmission and Execution of DC in the DCUG

When an authorized user in the DCUG requests DC, the DCUG manager transmits a partially encrypted DC in the cache scope to the user. Then, the user decrypts the received DC in the user's personal browser, and executes the DC using a player installed in the personal browser. Fig. 3 presents the procedure of the transmission of content from the DCUG to the DC.

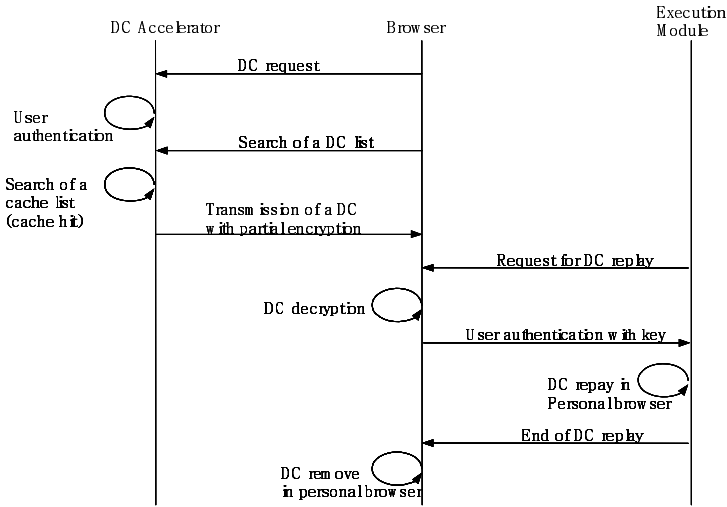


Fig. 3. Transmission and execution of DC in the DCUG

In order to execute DC in the DCUG, an exclusive browser, which has the function of opening the DC of the DCUG server, transmission of the personal information of the user, receiving DC, decryption, and play-back of DC, is required.

### 3 Analysis

Test results were compared to a frequently used existing commercial system, such as SecuMAX and Digicap, in order to verify the efficiency of the proposed system. The item of PEnc5% and PEnc7% present the 5% and 7% of partial encryption of DC, respectively. In addition, the item of Enc5%cache25 and PEnc5%cache40 present the processing speed tests of the 5% of partial encryption of DC at the cache-hit-ratio of 25% and 40%, respectively. The item of PEnc7%cache25 and PEnc7%cache40 present the processing speed tests of the 7% of partial encryption of DC at the cache-hit-ratio of 25% and 40%, respectively. Almost all commercial systems support a personal interface to assist the system security and user conveniences. As noted in Table 2, the level of security of DC was slightly reduced to improve the processing speed. This is due to the fact that the only application of encryption method and personal interface can't increase both the processing time of DC, and the level of security.

If web caching is not reflected in the system, the two existing systems present a more beneficial performance than that of the proposed system. However, the proposed system showed a high performance in the security of DC, and also presented an excellent processing speed from the aspect of considering a web caching. Because the numerical value of the test can be changed according to the test environment, it is not reasonable to conclude that the results present an absolute guideline to verify the system. However, the results revealed that the performance of the proposed system was improved compared to that of the existing commercial system.

**Table 2.** Analysis of the proposed system

Issue	Factors	Considerations	Approach of the proposed system
Processing Speed	Transmission speed of the network	Network traffics	Management for each DCUG group/layered structure web caching
	User execution speed	File size of the encryption/decryption	Layered structure system/partial encryption
Security	Security of the transmission	Safety	Public key method/management for each group
	Security of the execution	Speed lowering/ Reducing the execution process	Security of the DCUG /Certification for each DCUG group

The factors, which affect the processing speed of a digital content distribution system, are the delay according to the network traffic, and decryption process in user interfaces. The file size of the original sentence of DC increased due to the encryption. In addition, the encrypted transmission of a large amount of multimedia digital content, such as MP3, significantly increases the network traffic. The proposed system improves the processing speed by reducing these delay factors. The encrypted content in the DC server using a public key will be transmitted to the DCUG. The received DC can be decrypted using a personal key, and stored in a cache by applying a partial encryption. Finally, the authorized user of the DCUG will be supplied by DC, which is stored in a cache. Therefore, the traffic on the Internet for the user decreases, and the user will be affected by DC of the DCUG. In addition, because the user interface decrypts a partially encrypted content, the delay time to execute the content decreased.

The proposed system is secure, due to the fact that the DCUG, which has a personal key, can only decrypt the received DC. Because the user in the DCUG should be certified for each group, safety is guaranteed in the DCUG. In addition, the proposed system is secure enough to safely execute contents. The security of the DCUG can be guaranteed by the system itself. The authorized user of the DCUG, who is certified through the user certification, can only be allowed to access the cache list. It is necessary to make decryption when a user interface executes DC, and a certain additional security is guaranteed due to the fact that a single user, who has a proper key, can decrypt the DC.

The test results showed that the processing speed at the cache-hit-ratio of 25% was similar to that of the commercial system, and the processing speed was improved by 10%-18% at the cache-hit-ratio of 40%. Almost all commercial web caches present over 40% of the cache-hit-ratio. Thus, the test results revealed that the performance of the proposed system improved compared to the existing commercial system. In addition, it is possible to guarantee the security of DC without any decrease in the processing time.

## 4 Conclusions

This study designed a digital content distribution system, which can increase the execution speed, while guaranteeing the safety of DC. The proposed system introduced in this study reduces the delay factor, which is due to the network traffic during the execution of DC, using a layered web caching. In addition, this system uses a layered encryption/decryption to improve the level of security of DC. The test applied in this study compares the execution speed and level of security of the proposed system with the existing commercial system. As a result, an improvement in the level of security and execution speed of the proposed system was verified.

## References

- [1] Spectral Lines, "Talking About Digital Copyright," IEEE Spectrum, vol.38 Issue:6, pp.9, June 2001.
- [2] Thorwkrth N. J., Horvatic P., Weis R., Jian zhap, "Security methods for MP3 music delivery," Signals, Systems and Computers, 2000. Conference Record of the Thirty-Fourth Asilomar Conference on, vol.2, pp.1831-1835. 2000.
- [3] R. Rivest, A. Shamir and L. Adelman, "A Method for Obtaining Digital Signatures and Public Key Cryptosystems," Communications of the ACM, vol.21, No.2, 1978, pp.120-126
- [4] Korea Information Security Agency, A Development and Analysis Report on 12bits Block Encryption Algorithm(SEED), 1998.
- [5] Secumax: DRM Solution,(<http://www.secumax.com>)
- [6] M. Just, S. Vaudenay, "Authenticated Multi-Part Key Agreement", In Advances in Cryptology - ASIACRYPT'96 LNCS 1163, pp.36-49, 1996.
- [7] H. Bahn, S. Noh, S. L. Min, and K. Koh, "Efficient Replacement of Nonuniform Objects in Web Caches," IEEE Computer, Vol.35, No.6, pp.65-73, June 2002.
- [8] L. Rizzo, L. Vicisano, "Replacement Polices for a Proxy Cache," IEEE/ACM Trans. Networking, vol.8, no.2, pp.158-170, 2000.
- [9] C. Aggarwal, J. Wolf and P. Yu, "Caching on the World Wide Web," IEEE Trans. Knowledge and Data Engineering, vol.11, no.1, pp.94-107, 1999.



# Access Policy Sheet for Access Control in Fine-Grained XML

Jing Wu<sup>1</sup>, Yi Mu<sup>1</sup>, Jennifer Seberry<sup>1</sup>, and Chun Ruan<sup>2</sup>

<sup>1</sup> School of Information Technology and Computer Science,  
University of Wollongong, Wollongong, NSW 2522, Australia  
{jw91, ymu, jennie}@uow.edu.au

<sup>2</sup> School of Computing and IT, University of Western Sydney, Penrith, Australia  
chun@cit.uws.edu.au

**Abstract.** We propose an access control scheme for developing authorization rules for XML documents, allowing flexible data granularity and authorization propagation. To simplify the complex access control policies in XML, we introduce a new tool: *Authorization Policy Sheet* (APS). Complex access control rules can be easily described in an APS. The administrator of a system can easily manage the access control of the system. With aid of Data Type Definitions (DTD), the policies given in an APS can be converted into a standard XML code that can be implemented in a normal XML environment.

## 1 Introduction

Using eXtensible Markup Language has brought profound changes in the nature of information and the usability of the Web. XML can conveniently represent document structure and contents and offers great control over information granularity via transformation and query languages. As XML becomes a widespread data encoding format for Web applications, the Web resource protection must follow to withstand increasing threats from Internet hackers.

One important mechanism to protect Web contents is Access Control. An access control service is needed when some people want to block or allow access to an entire XML instance, while others would like to control access at the tag level. Developing an access control model, and related mechanism, in terms of XML is an important step. With the rapid development of web environments XML data access control has been intensively studied (e.g., [2, 8, 4, 5, 3, 1, 7]). However, these existing research works on XML do not offer more advanced access control features such as authorization delegation.

The popular mechanism in XML based access control takes advantage of Data Type Definitions (DTD) (e.g., [1]). With a DTD based approach, the access control rules are defined in DTD. A DTD based approach is sometimes employed along with a customized XML access control language where the rules are described using the language [6]. The merit of this kind of scheme is that the system administrator can enter/update access control rules much more conveniently.

In practice, access control components come with an order. For example, an order could be: `super_user > group_member > others`, where `super_user` holds the highest privilege and `others` hold the least privilege. In objects, an order can be associated with the depth the objects reside in a tree or a directory. Access control in such systems implies the propagation of authorization in terms of the associated order. In this paper, we present a novel framework for fine grain XML access control that includes delegation, in terms of the partial order of access control components. With our scheme, the complicated assess control task in XML documents becomes easy, since we propose a novel access control “spreadsheet” tool for describing rules. We call it *Authorization Policy Sheet* (APS). Using an APS with the associated DTD forms a normal XML code that is understandable to a normal XML environment.

The rest of the paper is organized as follows. In Section 2, we describe our access control model and define the major components in our model. In Section 3, we present the authorization policy sheet (APS), which forms the foundation to our model. In Section 4, we provide the definitions of predicates. The final section is the conclusion.

## 2 Basic Definitions

In this section, we define our access control model. We give the definitions of DTD, APS, and associated system components including subjects, objects, authorization rights, and types.

**Subject.** A subject is active. It could be a user or a processor. A subject has a name and other associated information dependent on the application. We require subjects to be either ordered with a proper order or unordered when the order of subjects are insignificant.

**Subject Set.** Subject constant poset  $(S, >)$ :  $admin, s_1, s_2, \dots, s_n$  denote ordered subjects with the order of  $admin > s_1 > s_2 > \dots > s_n$ . We assume that the administrator possesses the highest privilege.

A subject can be defined according to the need. For example, a subject could be described by set of attributes such as name, address, rights, etc. As the simplest example, in DTD, the subject is defined as:

```

<!DOCTYPE subject[
  <!ELEMENT   subject   (users*)>
  <!ELEMENT   users     (name)>
  <!ELEMENT   name      #PCDATA>
]>
```

The attribute to the above subject set contains only the usernames.

Here is the example of the subject hierarchy with three subjects (`Admin`, `Alice`, `Bob`), described in a separate sheet:

```

<subject>
  <users>
    <name> Admin </name>
  </users>
  <users>
    <name> Alice </name>
  </users>
  <users>
    <name> Bob </name>
  </users>
</subject>

```

We have omitted “partial order” of the associated subjects such as `Admin > Bob > Alice` as this will be presented later on.

**Object.** Objects are passive. They could be files, programs, tables, etc. Objects are represented by a constant poset  $(O, >)$ :  $o_1, o_2, \dots$  with the order  $o_1 > o_2, > \dots$ . The object is described as `target + path( $V, E$ )`, where `target` is an XML document or URL address, `path` is an XPath expression that eventually selects specific portions (object) of the XML document in the XML tree where  $V$  is a set of nodes and  $E$  is a set of edges. The structure of the objects could be defined in the DTD as follows.

<pre> &lt;!DOCTYPE  object[ &lt;!ELEMENT  object      (target,path)&gt; &lt;!ELEMENT  target      href #PCDATA&gt; &lt;!ELEMENT  path        #PCDATA&gt; ]&gt; </pre>
---

Here is the example of the object hierarchy described in a separate XML sheet:

```

<object>
  <object>
    <target> hospital.xml </target>
    <path>//doctor/operation_info</path>
  </object>
  <object>
    <target> hospital.xml </target>
    <path>//doctor/personnel_info/alice </path>
  </object>
</object>

```

**Access Rights.** Ordered rights are defined as constant poset  $(A, >)$ :  $a_1, a_2, \dots$  with the order:  $a_1 > a_2 > \dots$ . For example, `Read > Write > Executable`. They are defined in DTD as follows.

```

<!DOCTYPE access_right[
  <!ELEMENT access_right (a*) #IMPLIED>
]>

```

**Authorization Type.** Authorization type is given by the constant set  $T = \{n, p, d\}$ , where  $n$  is the negation w.r.t. an access right,  $p$  specifies grant w.r.t. an access right, and  $d$  specifies “delegable” w.r.t. an access right.

The ordered authorization types are described in DTD:

```

<!DOCTYPE authorization_type[
  <!ELEMENT authorization_type (n|p|d) #REQUIRED>
]>

```

### 3 Authorization Policy Sheet

Directly using XML to describe access control often shows little advantage when the access control system is complicated (e.g., when authorization delegation and propagation are required). In our model, authorization specifications or rules are provided in an Authorization Policy Sheet (APS) associated with the document/DTD. In APS, the representation of authorizations is described in terms of orders of the objects and subjects and explicit authorization rules.

The APS is separate from the document and DTD and offers great convenience in the administration of access control for system administrators due to its simplicity. The system administrator can manage the system access control by the concise rules given in an APS. The resultant XML sheet can be generated from the corresponding DTD and APS. APS also shows the great advantage due to its convenience in the specification of explicit rights and the implicit rights for XML documents.

#### 3.1 Rules

An APS sheet consists of a finite set of rules. A rule consists of *name*, *head* and *attribute*. When the head of a rule is an authorization predicate, the rule is called authorization rule. For a set of rules named  $r$ , each rule consists of a predicate and an attribute:

```

<rule:r>
  <p1, attribute> <- <condition>
  <p2, attribute> <- <condition>
  <p3, attribute> <- <condition>

  <pn, attribute> <- <condition>
</rule:r>

```

Here,  $p_1, p_2, \dots, p_n$  are a set of predicates and *attribute* denotes the components associated with the predicate. *condition* denotes the condition with

respect to the rule of a predicate. Due to the space limit, we will omit the details in this paper and will present it in the full version of the paper.

The structure of rule in DTD is defined as following:

```

<!DOCTYPE rule[
<!ELEMENT   rule      (predicate+,condition*)>
<!ELEMENT   predicate (grant, cangrant ) #IMPLIED>
<!ELEMENT   condition (gran, cangrant)*>
]>

```

A rule in XML is defined as following:

```

<rule:r>
  <predicate>
    <attribute>
      <attr>
    </attribute>
    ...
    <attribute>
      <attr>
    </attribute>
    <condition>
    ...
    </condition>
  </predicate>
  ...
</rule:r>

```

### 3.2 Partial Order

The partial orders of the access control components, including subjects, object, types, and rights, are one of the key components in an APS. We will see that they can be used to simplify our access control system by implicit rules in authorization propagations. In an APS, the partial orders are respectively defined in the form:  $(s_1 > s_2 > s_3 > \dots)$ ,  $(o_1 > o_2 > o_3 > \dots)$ ,  $(t_1 > t_2 > t_3 > \dots)$ ,  $(a_1 > a_2 > a_3 > \dots)$ .

## 4 Predicates

Predicates form the essential part of an APS. In our system, there is a set of predicates:  $P = \{p_1, p_2, \dots, p_n\}$ . Every predicate is constructed in the form  $\langle p_i, x_1, \dots, x_n \rangle$ .  $x_1, \dots, x_n$  are terms associated with the predicate. We utilize following predicates in an APS.

#### 4.1 grant

**Definition 1.** (*grant*) *grant* is a 6-tuple predicate  $S \times O \times T \times A \times S \times F$ :  $\langle grant, s, o, t, a, g, f \rangle$ , where subject  $s \in S$  is granted by grantor  $g \in G$  the access right  $a \in A$  on object  $o \in O$  with the type  $t \in T$ . It determines whether a subject is granted an access right over an object. In an APS, this rule reads:

```
<grant grantee=" ", target+path=" ", authorization_type=" ",
      access_right=" ", grantor=" ", status=" ">
```

Predicate *grant* in APS is an authorization rule, where the element *grant* has attributes *grantee*; *target + path*( $V, E$ ), *target* is an XML or DTD, *path* is an XPath expression that eventually selects specific portions (object) of the XML document in XML tree where  $V$  is a set of nodes and  $E$  is a set of edges, *authorization\_type*, *access\_rights*, *grantor*, and *status*. *status* is a flag indicating whether or not the rule is effective.

For example,

The merit of *grant* in ASP is obvious. The *grant* is also defined in DTD as follows.

```
<!DOCTYPE grant [
  <!ELEMENT grant (subject,object,authorization_type,
                  access_right,subject,status)>
  <!ELEMENT subject (grantee)>
  <!ELEMENT grantee (name)>
  <!ELEMENT name #PCDATA>

  <!ELEMENT object (target,path)>
  <!ELEMENT target href #PCDATA>
  <!ELEMENT path #PCDATA>

  <!ELEMENT authorization_type (n|p|d) #REQUIRED>
  <!ELEMENT access_right (a*) #IMPLIED>
  <!ELEMENT a #PCDATA>

  <!ELEMENT subject (grantor)>
  <!ELEMENT grantor (name)>
  <!ELEMENT name #PCDATA>

  <!ELEMENT status (true|false) #REQUIRED>
]>
```

The *grant* rule defined in ASP and DTD is converted into a standard form of XML. Here is an example of *grant* in APS and XML:

– rule in APS:

```
<rule:hospital>
  <grant, grantee="Alice",
    target+path="hospital_info.xml
                + //hospital/operation_info/",
    authorization_type="p",
    access_right="Read",
    grantor="Bob",
    status="True">
</rule:hospital>
```

– rule converted to XML:

```
<rule:hospital>
  <grant>
    <subject>
      <grantee>
        <name> Alice </name>
      </grantee>
    </subject>
    <object>
      <target>hospital_info.xml </target>
      <path>//hospital/operation_info</path>
    </object>
    <type> p </type>
    <access_right> Read </access_right>
    <subject>
      <grantor>
        <name> Bob </name>
      </grantor>
    </subject>
    <status> True </status>
  </grant>
</rule:hospital>
```

which reads that the grantee Alice is granted by the grantor Bob the access right read on Xpath specified object

hospital\_info.xml + //hospital/operation\_info/  
with the authorization type p and the status is set to True.

## 4.2 cangrant

Differing from `grant`, the predicate `cangrant` represents the capability of a subject in granting a right with respect to an object to another subject. Formally, we define it as follow.

**Definition 2.** (*cangrant*) *cangrant* is a 4-tuple predicate  $S \times O \times A \times F: \langle \text{cangrant}, s, o, a, f \rangle$  where  $s \in S$  is the grantor;  $o \in O = \text{target} + \text{path}(V, E)$ , *target* is an XML or DTD, *path* is an XPath expression that eventually selects specific portions (object) of the XML document in XML tree,  $V$  is a set of nodes and  $E$  is a set of edges;  $a \in A$  is an access right; and  $f \in F$  is the status of the rule. *cangrant* determines whether a subject can grant an access right over an object.

The definition above states that Subject  $s$  has the right to grant access right  $a$  on object  $o$  to other subjects.

In APS, we define the *cangrant* as

```
<cangrant subject=" ", target+path=" ", access_right=" ",
                                status=" ">
```

In DTD, *cangrant* is defined as follows:

```
<!DOCTYPE cangrant[
<!ELEMENT cangrant (subject,object, access_right,
                    status)>
<!ELEMENT subject (grantee)>
<!ELEMENT grantee (name)>
<!ELEMENT name #PCDATA>

<!ELEMENT object (target,path)>
<!ELEMENT target href #PCDATA>
<!ELEMENT path #PCDATA>

<!ELEMENT access_right (a*) #IMPLIED>

<!ELEMENT status (true|false) #REQUIRED >
]>
```

The following is an example of *cangrant* in XML.

– rule in APS:

```
<rule:hospital>
  <cangrant, subject="Alice",
            target+path="hospital_info.xml
                        + //hospital/operation_info",
            access_right="Read",
            status="True">
</rule:hospital>
```

– rule can be converted XML (omitted).

This rule states that Alice can grant the right read with respect to the object `//hospital_info + //hospital/operation_info` to other subjects.



### 4.3 Delegation

We define the *delegation* right  $d$ , which allows a subject who holds an access right to grant the right to another subject. A subject can grant other subjects an access right  $a$  over object  $o$  if the subject has an associated **cangrant** right and  $s$  is the security administrator **admin**, or  $s$  has been granted  $a$  over  $o$  with delegable type  $d$ . Clearly, the type  $d$  is a flag indicating whether or not the access right can be further granted to another subject by the holder of the access right.

We also assume that if subject  $s$  receives a delegable authorization directly or indirectly from another subject  $s'$  on some object  $o$  and access right  $a$ , then  $s$  cannot grant  $s'$  authorization on the same  $o$  and  $a$  later on.

For example, Alice is granted an access right **Read|Write** of type **p|d** over object `hospital_info.xml + //hospital/operation_info/`, then we have

– rule in APS:

```
<rule:hospital>
  <grant, grantee="Alice",
    target+path="hospital_info.xml
                + //hospital/operation_info/",
    authorization_type="p|d",
    access_right="Read|Write",
    grantor="Bob",
    status="True">
</rule:hospital>
```

This rule implies that Alice can grant the access right with respect to  $o$  and  $a$  to another subject. In other words, the list of rules can be updated whenever Alice takes the action.

With `rule:hospital` and **cangrant** right,

```
<cangrant,subject="Alice",
  target+path="hospital_info + *",
  access_right="Read",
  status="True">
```

A new rule can be generated by Alice (notice that **Write** access is forbidden):

```
<rule:hospital2>
  <grant, grantee="Cindy",
    target+path="hospital_info.xml
                + //hospital/operation_info/",
    authorization_type="p",
    access_right="Read",
    grantor="Alice",
    status="True">
</rule:hospital2>
```

where we have assumed that **d** denotes the non-inherited delegation. This means that the delegatee Bob cannot further delegate the right to other parties.

## 5 Conflict Resolution

Our model supports different types of access rights, authorizations can be in conflict where a user is granted two different types of access rights. Thus a proper conflict resolution policy is needed. We solve conflicts as follows. First, trace the delegation relation path explicitly. When a conflict occurs, we will see if the two grantors fall into a delegation path. If they do, then let the authorization with the grantor as the predecessor in the path override the other one. In other words, along the delegation path, the predecessors' grants have higher priority than the successors' grants. This policy can support well-controlled delegation. Second, if the conflicts can not be solved by the above policy, we will use Negative-takes-precedence based policy to resolve the conflicts. That is, we will resolve the conflicts according to their types, and the priority sequence is  $n > p > d$ . This policy favours security.

In Conclusion, we have presented an access control model that supports fine-grained XML. Our model has the following main merits. simplicity in access control management. Our access control model makes the complex access control management much more effective, simple and elegant. Due to lack of space, we will present the propagation of authorization in the full version of the paper.

## References

1. E. Bertino and E. Ferrari. Secure and selective dissemination of XML documents. *ACM Transaction on Information and System*, 5(3), 2002.
2. S. Castano, M. Fugini, G. Martella, and P. Samarati. *Database Security*. Addison-Wesley, 1995.
3. E. Damiani, S. D. C. di Vimercati, E. Fernandez-Medina, and P. Samarati. An access control system for svgdocuments. In *Proceedings of the Sixteenth Annual IFIPWG 11.3 Working Conference on Data and Application Security*, pages 121–135, 2000.
4. E. Damiani, S. Vimercati, S. Paraboschi, and P. Samarati. Design and implementation of an access processor for XML documents. In *Proceedings of the 9th international WWW conference*, pages 59–75, 2000.
5. E. Damiani, S. Vimercati, S. Paraboschi, and P. Samarati. Securing XML document. In *Proceedings of International Conference on Extending Database Technology (EDBT2000)*, pages 121–135, 2000.
6. A. Gabillon and E. Bruno. Regulating access to xml documents. In *Proceedings of the fifteenth annual working conference on Database and application security*, pages 299–314, 2001.
7. M. Kudo and S. Hada. XML document security based on provisional authorizations. In *Proceedings of the 7th ACM Conference on Computer and Communications Security*, pages 87–96, Athens, Greece, 2000.
8. C.-H. Lim, S. Park, and S. H. Son. Access control of XML documents considering update operations. In *Proceedings of ACM Workshop on XML Security*, pages 49–59, 2003.

# Monitoring the Health Condition of a Ubiquitous System: Rejuvenation vs. Recovery

Kazuki Iwamoto<sup>1</sup>, Tadashi Dohi<sup>1</sup>, and Naoto Kaio<sup>2</sup>

<sup>1</sup> Department of Information Engineering, Hiroshima University,  
Higashi-Hiroshima 739-8527, Japan  
dohi@rel.hiroshima-u.ac.jp

<sup>2</sup> Department of Economic Informatics, Hiroshima Shudo University,  
Hiroshima 731-3195, Japan  
kaio@shudo-u.ac.jp

**Abstract.** Software rejuvenation is a preventive and proactive solution that is particularly useful for counteracting the phenomenon of software aging. In this article, we consider a periodic software rejuvenation model based on the steady-state system availability in discrete operational circumstance. More precisely, we treat a telecommunication billing application as a simple ubiquitous application, and describe its stochastic behavior by applying the discrete renewal reward process. The main objective is the determination of the optimal frequency to rejuvenate the ubiquitous application, maximizing the steady-state system availability. Also, we develop a statistically non-parametric algorithm to estimate the optimal rejuvenation schedule with the discrete total time on test concept.

## 1 Introduction

Since ubiquitous systems around us are controlled by software in almost all cases, it is quite important to check their health condition occasionally or periodically. Software rejuvenation is a preventive and proactive solution that is particularly useful for counteracting the phenomenon of software aging. The faults involved in embedded software systems should ideally have been removed during the debugging phase. Even if the embedded software may have been thoroughly tested, it still may have some design faults that are yet to be revealed. Such design faults are called *bohrbugs* and may exist even in mature software such as commercial operating systems. Also, even mature software can be expected to have what are known as *heisenbugs*. These are bugs in the software that are revealed only during specific collusions of events. For example, a sequence of operations may leave the software in a state that results in an error on an operation executed next. Simply retrying a failed operation, or if the application process has crashed, restarting the process might resolve such a problem. Another type of fault observed in ubiquitous applications is due to the phenomenon of resource exhaustion. These faults may exist in not only middleware under ubiquitous computing circumstance but also typical operating systems and telecommunication applications. For instance, operating system resources such as swap space and free memory available are progressively depleted due to defects in software such as memory leaks and incomplete cleanup of resources after use.

In fact, when a ubiquitous application executes continuously for long periods of time, some of the faults cause the software to age due to the error conditions that accrue with time and/or load. *Software aging* will affect the performance of the application and eventually cause it to fail [1], [3]. A complementary approach to handle software aging and its related transient software failures, called *software rejuvenation*, are becoming popular [7]. Software rejuvenation is a preventive and proactive solution that is particularly useful for counteracting the phenomenon of software aging. It involves stopping the running software occasionally, cleaning its internal state and restarting it. Cleaning the internal state of a software might involve garbage collection, flushing operating system kernel tables, reinitializing internal data structures, and hardware reboot. For many ubiquitous applications, the rejuvenation may correspond to a system reconfiguration in the sleep mode during the idle period.

Huang *et al.* [7] report the software aging phenomenon in a real telecommunication billing application where over time the application experiences a crash or a hang failure, and propose to perform the rejuvenation occasionally. More specifically, they consider the degradation as a two step process. From the clean state the software system jumps into a degraded state from which two actions are possible: rejuvenation with return to the clean state or transition to the complete failure state. They model the four-state process as a continuous-time Markov chain, and derive the steady-state system availability. Avritzer and Weyuker [2] discuss the aging in a telecommunication switching software where the effect manifests as gradual performance degradation. Garg *et al.* [6] introduce the idea of periodic rejuvenation (deterministic interval between successive rejuvenations) into the Huang *et al.* model [7] and represent the stochastic behavior by using a Markov regenerative stochastic Petri net. Recently, Dohi *et al.* [4], [8] extend both the original Huang *et al.* model [7] and Garg *et al.* model [6] to semi-Markov models, and develop statistically non-parametric algorithms to estimate the optimal software rejuvenation schedules from the complete sample data of failure time.

It should be noted in the past literature that the software system is assumed to operate in continuous time. However, in many practical situations, system lifetime (degradation time) can not be measured in calendar time. For instance, consider the system failure of an embedded system where the number of rounds for use before system failure is more significant than the system age measured by CPU time. Also, in some mission critical systems based on the transaction processing, the number of transactions strongly depends on the software aging and its related transient failure. In such systems, the system lifetime should be regarded as a discrete random variable. Unfortunately, enough research has not been carried out on discrete software rejuvenation models. Dohi *et al.* [5] reformulate the semi-Markov rejuvenation models in [4] as those in discrete time under the steady-state system availability criterion. More specifically, they model the stochastic behavior of a telecommunication billing application discussed in [4], [7] by using discrete semi-Markov processes, and determine the optimal periodic software rejuvenation schedules in discrete-time setting. Similar to the paper [4], the same authors develop non-parametric estimators of the optimal

non-periodic software rejuvenation schedules maximizing the steady-state system availability [5].

This paper is a continuation of the above work [5]. That is, we reformulate the periodic software rejuvenation model in continuous time [6], [8] as a discrete one. In a fashion similar to the previous work [5], we apply the discrete Markov regenerative process and derive analytically the optimal periodic software rejuvenation schedule in discrete-time setting, maximizing the steady-state system availability. Also, we wish to emphasize here that the discrete model considered here is not a simple analogy of the continuous-time software rejuvenation model. First, we utilize the discrete total time on test (DTTT) concept (scaled DTTT transform and scaled DTTT statistics), which is first introduced in [5] and is a quite new statistical device. It is needed to develop a statistically non-parametric algorithm of the optimal software rejuvenation schedule. In addition, the results in [5] are based on the discrete-time semi-Markov processes, but our model considered in this paper is classified into the discrete-time Markov regenerative process which belongs to the more wide class stochastic process. As pointed out in the literature [6], [8], the periodic rejuvenation scheme is feasible in practice since the time interval between successive rejuvenations is deterministic. Also, it is noted that the resulting optimal rejuvenation schedule based on the discrete modeling framework is not always equivalent to the discretized one for continuous-time model, because the discrete model involves a delicate problem on the uniqueness of the optimal schedule. In that sense, our modeling approach developed in this paper is not a trivial example.

## 2 Model Description

### 2.1 Notation and Assumption

$Z$ : time interval from highly robust state to failure probable state (discrete random variable)

$F_0(n)$ ,  $f_0(n)$ ,  $\mu_0$  ( $> 0$ ): cumulative distribution function (cdf), probability mass function (pmf) and mean of  $Z$ , where  $n = 0, 1, 2, \dots$

$X$ : failure time from failure probable state (discrete random variable)

$F_f(n)$ ,  $f_f(n)$ ,  $\mu_f$  ( $> 0$ ): cdf, pmf and mean of  $X$

' \* ': discrete convolution operator, *i.e.*  $F_0 * F_f(n) = \sum_{j=0}^n F_0(n-j)f_f(j) = \sum_{j=0}^n F_f(n-j)f_0(j)$

$\bar{\psi}(\cdot)$ : survivor function ( $= 1 - \psi(\cdot)$ )

$r_{0f}(n)$ : failure rate ( $= f_0 * f_f(n) / \overline{F_0 * F_f(n-1)}$ )

$Y$ : recovery time from failure state (discrete random variable)

$F_a(n)$ ,  $f_a(n)$ ,  $\mu_a$  ( $> 0$ ): cdf, pmf and mean of  $Y$

$N$ : rejuvenation time from highly robust state (discrete random variable)

$F(n)$ ,  $n_0$  ( $\geq 0$ ): cdf and mean of  $N$

$R$ : system overhead incurred by software rejuvenation (discrete random variable)

$F_c(n)$ ,  $f_c(n)$ ,  $\mu_c$  ( $> 0$ ): cdf, pmf and mean of  $R$

**Assumption:**  $\mu_a > \mu_c$ , *i.e.* the mean recovery time from the system failure is strictly greater than the mean time to complete the rejuvenation.

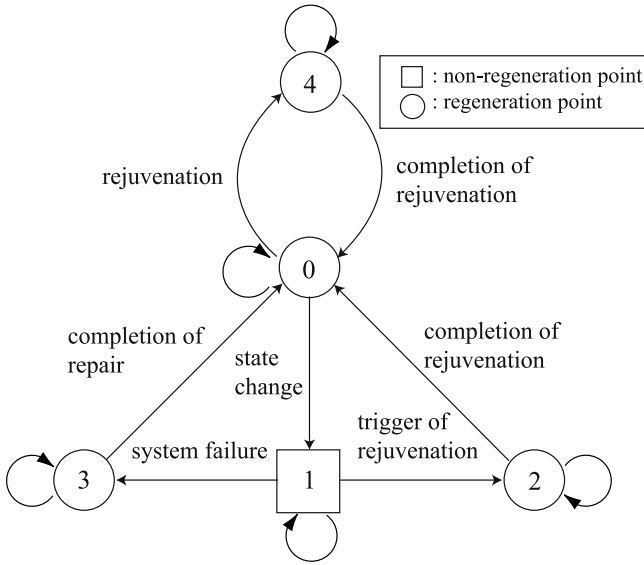


Fig. 1. Transition diagram

Consider the similar stochastic model with periodic software rejuvenation to Garg *et al.* [6] and Suzuki *et al.* [8] in discrete time. Suppose that the software system starts for operation at time  $n = 0$  and is in the highly robust state (normal operation state). Let  $Z$  be the random time to reach the failure probable state from the highly robust state. Let  $\Pr\{Z \leq n\} = F_0(n)$ , ( $n = 0, 1, 2, \dots$ ). Just after the state becomes the failure probable state, a system failure may occur with positive probability. Let  $X$  be the time to failure from the failure probable state having the cdf  $\Pr\{X \leq n\} = F_f(n)$ . Then, two actions are possible to be taken; rejuvenation and recovery from the system failure. If the system failure occurs before triggering a software rejuvenation, then the recovery operation starts immediately. The time to complete the recovery operation  $Y$  is also the positive random variable having the cdf  $\Pr\{Y \leq n\} = F_a(n)$ . Without any loss of generality, it is assumed that after completing recovery operation the software system becomes as good as new.

On the other hand, rejuvenation is performed at a random time interval measured from the start (or restart) of the software in the robust state. The cdf of the time to invoke the software rejuvenation, say  $N$ , and the cdf of the time to complete software rejuvenation are represented by  $F(n)$  and  $F_c(n)$ , respectively. Suppose that the time to rejuvenate the software is a constant  $n_0$ , *i.e.* the software rejuvenation is performed periodically. Then, the cdf  $F(n)$  has to be replaced by

$$F(n) = U(n - n_0) = \begin{cases} 1 : (n \geq n_0) \\ 0 : (n < n_0), \end{cases} \tag{1}$$

where  $U(\cdot)$  is the unit step function. We call  $n_0 (\geq 0)$  *the software rejuvenative schedule* in this paper. After completing the software rejuvenation, the software system becomes as good as new, and the software age is initiated at the beginning of the next highly robust state.

Define the following five states:

- State 0:** highly robust state (normal operation state)
- State 1:** failure probable state
- State 2:** software rejuvenation state from failure probable state
- State 3:** failure state
- State 4:** software rejuvenation state from highly robust state.

Figure 1 depicts the transition diagram for the stochastic model under consideration, where the states denoted by circles (0, 2, 3, 4) and square (1) are regeneration points and non-regeneration points, respectively, in the Markov regenerative process. Strictly speaking, this stochastic process is not a simple semi-Markov process. However, since only one non-regeneration point is included in the transition diagram, it can be reduced to an equivalent semi-Markov process.

### 3 Availability Analysis

We formulate the steady-state system availability as a criterion of optimality. Define the time length from the beginning of the system operation to the completion of the preventive or corrective maintenance of software system as one cycle. Suppose that the same cycle is repeated again and again over an infinite time horizon. The mean operative time during one cycle is obtained as

$$S(n_0) = \sum_{n=0}^{n_0-1} \overline{F_f * F_0}(n). \tag{2}$$

The mean time length of one cycle is given by

$$T(n_0) = \mu_c + (\mu_a - \mu_c)F_f * F_0(n_0) + \sum_{n=0}^{n_0-1} \overline{F_f * F_0}(n). \tag{3}$$

Then the steady-state system availability is, from the discrete renewal reward theorem, derived by  $AV(n_0) = S(n_0)/T(n_0)$ , and the problem is to seek the optimal software rejuvenation schedule  $n_0^*$  maximizing  $AV(n_0)$ .

Taking the difference of  $AV(n_0)$  with respect to  $n_0$ , define the following function:

$$q(n_0) = \frac{T(n_0 + 1)T(n_0)\{AV(n_0 + 1) - AV(n_0)\}}{F_0 * F_f(n_0)}$$

$$= (\mu_c - \mu_a)S(n_0)r_{0f}(n_0 + 1) + T(n_0) - S(n_0), \tag{4}$$

where  $q(0) = T(0) - S(0) + (\mu_c - \mu_a)S(0)r_{0f}(1) = \mu_c > 0$ . The following result gives the optimal software rejuvenation schedule.

**Theorem 1:** (1) Suppose that the convoluted cdf  $F_0 * F_f(n)$  is strictly IFR (increasing failure rate) under the assumption  $\mu_a > \mu_c$ .

(i) If  $q(\infty) < 0$ , then there exist (at least one, at most two) optimal software rejuvenation schedule  $n_0^*$  ( $0 < n_0^* < \infty$ ) satisfying  $q(n_0^* - 1) > 0$  and  $q(n_0^*) \leq 0$ . Then, the maximum system availability is given by

$$\underline{AV}(n_0^*) \leq AV(n_0^*) < \overline{AV}(n_0^*), \tag{5}$$

where

$$\underline{AV}(n_0^*) = \frac{1}{1 + (\mu_a - \mu_c)r_{0f}(n_0^* + 1)}, \tag{6}$$

$$\overline{AV}(n_0^*) = \frac{1}{1 + (\mu_a - \mu_c)r_{0f}(n_0^*)}. \tag{7}$$

(ii) If  $q(\infty) \geq 0$ , then the optimal software rejuvenation schedule becomes  $n_0^* \rightarrow \infty$ , *i.e.* it is optimal not to carry out the software rejuvenation. Then, the maximum system availability is given by  $AV(\infty) = \mu_f / (\mu_a + \mu_f)$ .

(2) Suppose that the convoluted cdf  $F_0 * F_f(n)$  is DFR (decreasing failure rate) under the assumption  $\mu_a > \mu_c$ . Then, the steady-state system availability  $AV(n_0)$  is a convex function of  $n_0$ , and the optimal software rejuvenation schedule is  $n_0^* = 0$  or  $n_0^* \rightarrow \infty$ .

In the following section, we give an alternative graphical interpretation of Theorem 1 by applying the DTTT concept.

### 4 Statistical Estimation Algorithm

For the discrete cdf  $F_f * F_0(n)$ , define the scaled DTTT transform [5]:

$$\phi(p) = \sum_{n=0}^{(F_f * F_0)^{-1}(p)} \frac{\overline{F_f * F_0}(n)}{\mu_f + \mu_0}, \tag{8}$$

where

$$(F_f * F_0)^{-1}(p) = \min\{n : F_f * F_0(n) > p\} - 1, \tag{9}$$

if the inverse function exists. Then it is evident that

$$\mu_f + \mu_0 = \sum_{n=0}^{\infty} \overline{F_f * F_0}(n). \tag{10}$$



The scaled DTTT transform can be regarded as a discrete analogy of the familiar total time on test transform of continuous cdf, and can be used to examine the aging property of discrete probability distribution. To our best knowledge, this interesting statistical device has not sufficiently been studied in the context of reliability statistics.

Based on the definition in Eq.(8), let us represent the system availability by the parameter  $p$  instead of  $n_0$ . Suppose that  $p$  is one-to-one corresponding to  $n_0$ . After a few algebraic manipulations, we can obtain the following result:

**Theorem 2:** Obtaining the optimal software rejuvenation schedule  $n_0^*$  maximizing the steady-state system availability  $AV(n_0)$  is equivalent to obtaining  $p^*$  ( $0 \leq p^* \leq 1$ ) such as

$$\max_{0 \leq p \leq 1} \frac{\phi(p)}{p + \beta}, \tag{11}$$

where  $\beta = \mu_c / (\mu_a - \mu_c)$ .

Theorem 2 is the dual of Theorem 1. From this result, it is seen that the optimal software rejuvenation schedule  $n_0^* = (F_f * F_0)^{-1}(p^*)$  is determined by calculating the optimal point  $p^*$  ( $0 \leq p^* \leq 1$ ) maximizing the tangent slope from the point  $(-\beta, 0)$  to the curve  $(p, \phi(p)) \in [0, 1] \times [0, 1]$  in the two-dimensional plane. This graphical idea is very useful to calculate the optimal rejuvenation schedule in practice. For the continuous-time models [4], [8], the optimal solution can be characterized as a unique solution of non-linear equation in the non-trivial case. On the other hand, in discrete-time setting, we have to solve simultaneous inequalities  $q(n_0^* - 1) > 0$  and  $q(n_0^*) \leq 0$ . If  $n_0$  is large, this is equivalent to solve a non-trivial combinatorial problem. However, from Theorem 2, we can plot the graph  $(p, \phi(p))$  easily and seek the optimal point  $p^*$  so as to maximize the tangent slope geometrically. The other benefit of this approach arises in an educational aspect to perform the sensitivity analysis. For varying model parameters, we can check the sensitivity of model parameters on the optimal rejuvenation schedule on the graph.

Next, suppose that the optimal software rejuvenation schedule has to be estimated from  $k$  ordered complete observations:  $0 = x_0 \leq x_1 \leq x_2 \leq \dots \leq x_k$  of the times from a discrete cdf  $F_0 * F_f(n)$ , which is unknown. Then, the empirical distribution for this sample, is given by

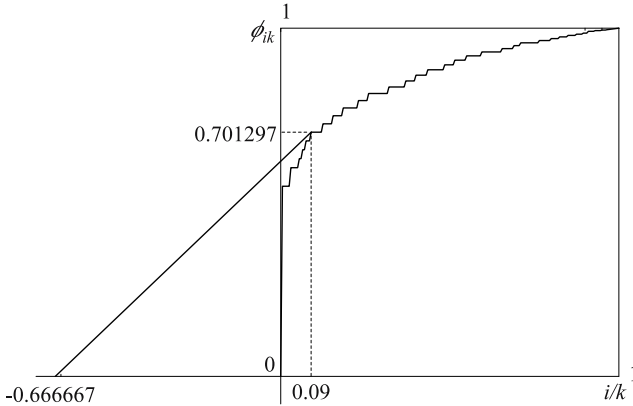
$$F_{fk}(n) = \begin{cases} i/k & \text{for } x_i \leq n < x_{i+1}, \\ 1 & \text{for } x_k \leq n. \end{cases} \tag{12}$$

The numerical counterpart of the scaled DTTT transform, called *scaled DTTT statistics*, based on this sample, is defined by

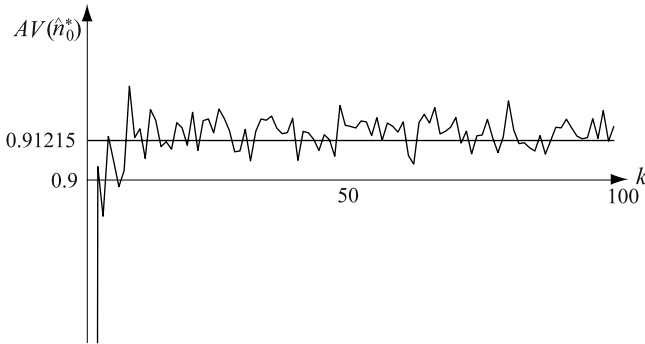
$$\phi_{ik} = \psi_i / \psi_k, \quad i = 0, 1, 2, \dots, k, \tag{13}$$

where

$$\psi_i = \sum_{j=1}^i (k - j + 1)(x_j - x_{j-1}), \quad i = 1, 2, \dots, k \tag{14}$$



**Fig. 2.** Estimation of the optimal software rejuvenation schedule



**Fig. 3.** Asymptotic behavior of the estimates for the maximum system availability

with  $\psi_0 = 0$ . The resulting step function by plotting the points  $(i/k, \phi_{ik})$  ( $i = 0, 1, 2, \dots, k$ ) is called *scaled DTTT plot*.

The following theorem gives a statistically non-parametric estimation algorithm for the optimal software rejuvenation schedule.

**Theorem 3:** Suppose that the optimal software rejuvenation schedule has to be estimated from  $k$  ordered complete sample  $0 = x_0 \leq x_1 \leq x_2 \leq \dots \leq x_k$  of the times from a discrete cdf  $F_f * F_0(n)$ , which is unknown. Then, a non-parametric estimator of the optimal software rejuvenation schedule  $\hat{n}_0^*$  which maximizes  $AV(n_0)$  is given by  $x_{j^*}$ , where

$$j^* = \left\{ j \mid \max_{0 \leq j \leq k} \frac{\phi_{jk}}{j/k + \beta} \right\}. \tag{15}$$

The above result is a direct application of Theorem 2. In fact, it can be expected that the resulting estimator  $x_{j^*}$  may function well to estimate the optimal soft-

ware rejuvenation schedule maximizing the steady-state system availability, if the number of sample is sufficiently large. Unfortunately, the strong consistency of  $\phi_{ik}$  has not been analytically proved in the literature, *i.e.*  $\phi_{ik} \rightarrow \phi(p)$  as  $k \rightarrow \infty$ . However, this hypothesis seems to be correct in our simulation experiments. In the following section, we give a numerical example to show usefulness of the estimation algorithm.

## 5 A Numerical Example

We present an example to determine the optimal software rejuvenation schedule which maximizes the steady-state system availability. Suppose that the time  $Z + X$  obeys the negative binomial distribution with pmf:

$$f_f(n) = \binom{n-1}{r-1} q^r (1-q)^{n-r}, \quad n = 1, 2, 3, \dots, \quad (16)$$

where  $q \in (0, 1)$  and  $r = 1, 2, \dots$  is the natural number. Also, it is assumed that  $Z$  is a geometrically distributed random variable having pmf:

$$f_0(n) = p(1-p)^n, \quad n = 0, 1, 2, \dots \quad (17)$$

Suppose that  $(r, q) = (10, 0.3)$ ,  $p = 0.3$ ,  $\mu_a = 5.0$  [day] and  $\mu_c = 2.0$  [day]. In this case, the optimal software rejuvenation schedule is given by  $n_0^* = (F_f * F_0)^{-1}(0.125291) = 26$  with  $AV(26) = 0.912150$ , if we can know the pmf  $f_f(n)$  completely. In Fig. 2 we show the estimation result of the optimal software rejuvenation schedule, where the failure time data are generated from the negative binomial distribution. For 200 simulation data (negative binomial distributed random number), the estimates of the optimal periodic rejuvenation schedule and its associated system availability are given by  $\hat{n}_0^* = x_{19} = 26$  and  $AV(\hat{n}_0^*) = 0.918798$ , respectively.

Of our next concern is the investigation of asymptotic property of the estimators given in Theorem 3. In the simulation experiment, we generate the negative binomial distributed random number as the system failure time data and sequentially estimate the optimal software rejuvenation schedule and the corresponding system availability. In Fig. 3, we plot estimates of the maximum system availability, where the horizontal line denotes the really maximum availability  $AV(26)$  mentioned before. From this figure, it is observed that the estimate of the steady-state system availability fluctuates around the real maximum and that the non-parametric method proposed here can provide a good estimate. As the number of data increases, the estimated system availability converges to the corresponding asymptotic value. Hence, the estimator of the optimal software rejuvenation schedule can be expected to be consistent numerically. On the other hand, from this result, it can be seen that a number of system failure data are not always needed to get the good estimates with higher accuracy. For instance, in the above example, around 30 data will be enough even if we can not know the underlying system failure time distribution. This

result tells us that the proposed statistical estimation method provides a good predictive performance of the optimal rejuvenation schedule for an operational software system.

## Acknowledgments

The present research was partially supported by a Grant-in-Aid for Scientific Research from the Ministry of Education, Sports, Science and Culture of Japan under Grant Nos. 15651076 and 16310116, the Research Program 2005 under the Institute for Advanced Studies of the Hiroshima Shudo University.

## References

1. Adams, E. (1984), Optimizing preventive service of the software products, *IBM J. Research & Development*, **28**, 2–14.
2. Avritzer, A. and Weyuker, E. J. (1997), Monitoring smoothly degrading systems for increased dependability, *Empirical Software Eng.*, **2**, 59–77.
3. Castelli, V., Harper, R. E., Heidelberger, P., Hunter, S. W., Trivedi, K. S., Vaidyanathan, K. V. and Zeggert, W. P. (2001), Proactive management of software aging, *IBM J. Research & Development*, **45**, 311–332.
4. Dohi, T., Goševa-Popstojanova, K. and Trivedi, K. S. (2001), Estimating software rejuvenation schedule in high assurance systems, *The Computer Journal*, **44**, 473–485.
5. Dohi, T., Iwamoto, K., Okamura, H. and Kaio, N. (2003), Discrete availability models to rejuvenate a telecommunication billing application, *IEICE Trans. on Communications (B)*, **E86-B**, 2931–2939.
6. Garg, S., Telek, M., Puliafito, A. and Trivedi, K. S. (1995), Analysis of software rejuvenation using Markov regenerative stochastic Petri net, *Proc. 6th Int'l Symp. on Software Reliab. Eng.*, 24–27, IEEE CS Press.
7. Huang, Y., Kintala, C., Koletti, N. and Fulton, N. D. (1995), Software rejuvenation: analysis, module and applications, *Proc. 25th Int'l Symp. on Fault Tolerant Computing*, 381–390, IEEE CS Press.
8. Suzuki, H., Dohi, T., Goševa-Popstojanova, K. and Trivedi, K. S. (2002), Analysis of multistep failure models with periodic software rejuvenation, *Advances in Stochastic Modelling* (J. R. Artalejo and A. Krishnamoorthy, eds.), Notable Publications, Inc., 85–108.

# A Dependability Management Mechanism for Ubiquitous Computing Systems\*

Changyeol Choi and Sungsoo Kim

Graduate School of Information and Communication, Ajou University, Suwon, Korea  
{clchoi, sskim}@ajou.ac.kr

**Abstract.** Dependability for a ubiquitous computing system must be guaranteed for each single component of a system and for the whole system, which—designed to fulfill a certain task—might be more than just a sum of its components. Ubiquitous computing systems must enable the testing not only each of software components separately but also of the whole system. In this paper, we propose a management mechanism for applying software rejuvenation technology into a ubiquitous computing system. It adopts the automatic monitoring scheme, the automatic analysis scheme, the autonomic plan and the execution scheme to suggest the optimal configuration alternative of a ubiquitous computing system. We validate the autonomic fault management scheme based on a workload model derived from the system log analysis.

## 1 Introduction

Computing has spread from a desktop computer to several areas such as automobiles, gadgets, telecommunications and the Internet; that is, hundreds of internet-enabled computers per human being, none of them resembling a conventional keyboard-and-monitor machine. Particularly, a ubiquitous computing system could be constructed not only with one computer but with networks of computers (or other devices with computing power) embedded in everyday objects where computers are made available throughout the physical environment [1] and must be able to operate for 24 hours/7 days with the minimum human intervention. For this, dependability must be guaranteed for each single component of a system and for the whole system, which—designed to fulfill a certain task—might be more than just a sum of its components. In order to deal with faults efficiently, modular and structured systems—enable loosely coupled, componentized systems, such as Sun J2EE[2] and Micro soft .NET[3]—are mostly preferable. However, when a certain number of components fail owing to unpredicted failures, the system will fail to guarantee service quality to a large number of users. In addition to hardware failures, many unexpected system faults are caused by software failures [4,5,6].

Software-aging phenomenon, a dormant form of software faults, can lead to data loss, communication disruption, and can be aggravated by malfunction such as memory leak, buffer overflow, data corruption, or numerical error accumulation, etc.

---

\* This research is supported by the Ubiquitous Autonomic Computing and Network Project, the Ministry of Information and Communication (MIC) 21st Century Frontier R&D Program in Korea.

Software rejuvenation is based on the idea of preventive maintenance applied to the software context that is widely employed in different engineering disciplines [7,8,9,10]. Software rejuvenation is a proactive fault management technique aimed at cleaning up the system internal state to prevent the occurrence of more severe crash failures in the future. It involves occasionally terminating an application or a system, cleaning its internal state and restarting it. Garbage collection, flushing operating system kernel tables, and reinitializing internal data structures are some examples of what cleaning the internal state of software might involve. Recent studies of the software rejuvenation technique stress the importance of the analysis on availability, downtime cost, and other methodology to improve system availability. However, it is necessary to address the management complexity issues by using technology to manage the software rejuvenation technology with the minimum human intervention. Moreover, they must enable the testing not only each of software components separately but also of the whole system. So, in this paper, we propose a management mechanism for applying software rejuvenation technology into a ubiquitous computing system. It adopts the automatic monitoring scheme to collect, aggregate, filter, and track related metrics, the automatic analysis scheme to extend our previous work[10], the autonomic plan and execution scheme to suggest the optimal configuration alternative of a ubiquitous computing system. We validate the autonomic fault management scheme based on a workload model derived from the system log analysis. The rest of this paper is organized as follows. The autonomic fault management mechanisms are presented in section 2. Section 3 describes how to get the metrics, analyze the availability, and make the suggestion for the dependability requirements in detail. Section 4 presents the analytic and simulation results based on experiments on different system parameters. Section 5 concludes the paper.

## 2 Autonomic Fault Management Mechanisms

The CIM(common information model) standards promoted by DMTF (distributed management task force) are a way to systematize the available information about the computing environment in which semantically rich management information between systems could be exchanged [11]. The approach uses a uniform modeling formalism that, together with the basic scenario of object-oriented constructs, supports the cooperative development of an object-oriented schema across multiple organizations. So, we describe with CIM and UML (Unified Modeling Language) the autonomic fault managements for a ubiquitous computing system—divided into two parts: the dependability manger for the plan and execution and the observation manager for the monitor and analysis (refer in Figure 1).

A class represents a concept within the system being modeled are denoted by the rectangular boxes in the diagram of Fig. 1. Classes contain the class name, the point data structure that has been defined for the class, and the behavior that can be performed on instances of this class. Multiplicity represents the number of instances of each class that form part of the relation with the lines connecting the classes, for examples, “a Dependability Manager controls one or many Observation Manager; a Observation Manager is managed by one Dependability Manager.” A brief description of the classes in alphabetical order follows for the dependability manager and the observation manager, respectively.

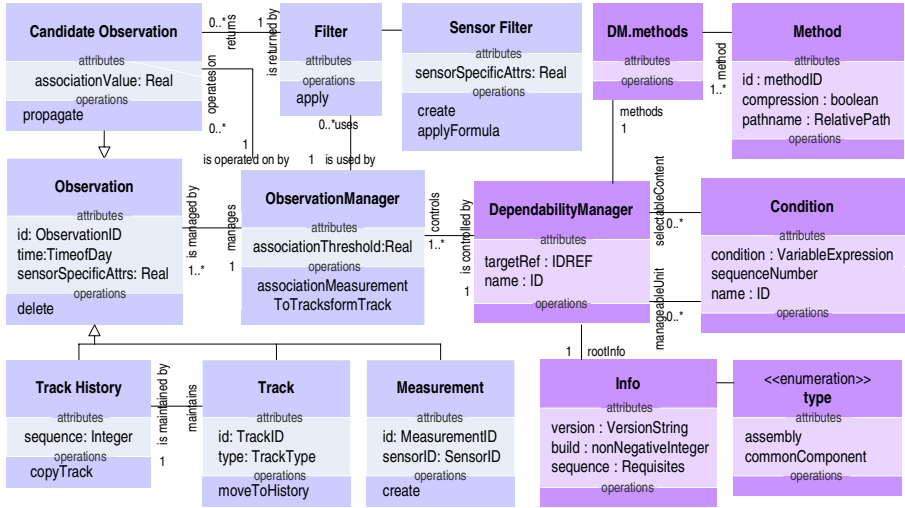


Fig. 1. UML representation of autonomic fault management mechanisms

### 2.1 Dependability Manager

Dependability Manager (DM) plays a core role in the autonomic fault management in order to apply software rejuvenation technology into a ubiquitous computing system. It interprets the quality of applications and systems against a predefined service level, and finally invokes methods to correct a problem if necessary. In order to make a plan and take a countermeasure, DM cooperates with the observation manager that analyzes the properties of objects aggregated in the model and generates indications about the status of those resources.

- **Condition**: the generic type describing any contained component defined within a ubiquitous computing system. The condition is a boolean variable expression. When the condition is false, the DM invokes any method.
- **Method**: a composite of action definitions including a method identifier. Some of the actions defined in a method may specify the execution of custom code with the path in order to satisfy some given requirements.
- **Info**: the definition for describing the characteristics of the system information installed on the version and the requisite target to satisfy a software requirement.

### 2.2 Observation Manager

The observation manager adopts the automatic monitoring and analysis scheme to collect, aggregate, filter, track related metrics, and finally generate indications about the status of a ubiquitous computing system.

- **Filter**: base class for all sensor filters. Application of the filter will then identify the tracks or measurements that lie within the volume determined by the filter.
- **Measurement**: all measurements provided by a sensor or communications interface and received by the Observation Manager. The attributes in this class have not been fully defined.

- **Observation:** object data managed within the fault management domain. Candidate observation is created for each track or measurement that matches the filter criteria.
- **Track:** State data about a track object. The attributes in this class have not been fully defined and record the history associated to a specific track.

### 3 Automation Scheme

We divide the hazards of a ubiquitous computing system into three errors: environment and hardware structure error, system error, human error (refer in Figure 2). The system comes to an emergency stop if it fails to detect, prevent, or recover at least one of the three errors. Because many unexpected system faults are caused by software failures, we concentrate on system error in operation—sensor error, controller error, and actuator error. For automatic monitor, the reference model for a resource, component, or system must be constructed. The observation reference model controls the data gathering and interprets the data according to best-practices algorithms in order to detect the unpredicted fault phenomenon condition. For an example, we build up the observation reference model for a memory leakage phenomenon.

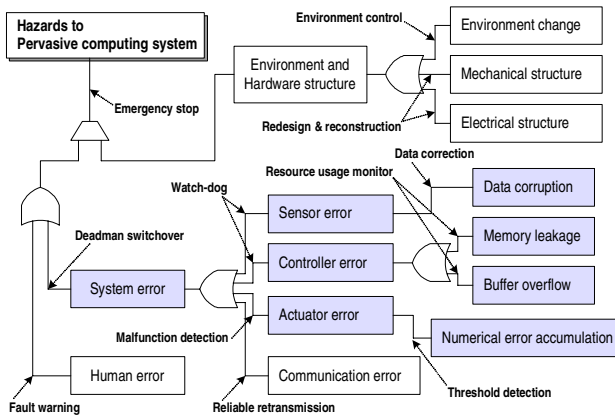


Fig. 2. Fault classification and possible events

A memory leakage occurs, when it is short of available memory with high working set and deficient in memory for private bytes of a specific process used in a software component. This phenomenon can be detected by tracking the state change of objects such as memory and process from the past state[12]. That is, the number of bytes used for the working set of the process, the number of bytes committed to a process, the total number of page faults caused by a process or the total memory available might be examined. If the portion of total working set in memory exceeds one of available memory, the shortage of memory with high working set occurs. If the current number of process private bytes exceeds the previous value of them accidentally, then the deficiency of private memory is observed, too. However, only quantitative monitoring and analysis is incomplete for rigorous fault management. It is necessary to adopt best-



practices fault injection algorithm[13,14] for obtaining the failure rate of a software object (application, component, service, or interface) because the software aging phenomenon would occur in long mission time. Thus, the dependability manager has the capability to evaluate the robustness of its software object for itself. In addition to the functioning software to perform system checking, loader, applications, etc. an important design issue for the ubiquitous computing system is related to the number of components that should be placed in different operational modes: operations, repair, dormant, etc. The serviceable number of components may decline because of software aging phenomenon after a long mission and then, the number of components in dormant mode must be incremented to that degree. A component having dormant software failure can switch to operation mode by software rejuvenation or will come down to system failure. In the case of failure, that component must be repaired. A fault-repair model is necessary to carry out the analysis.

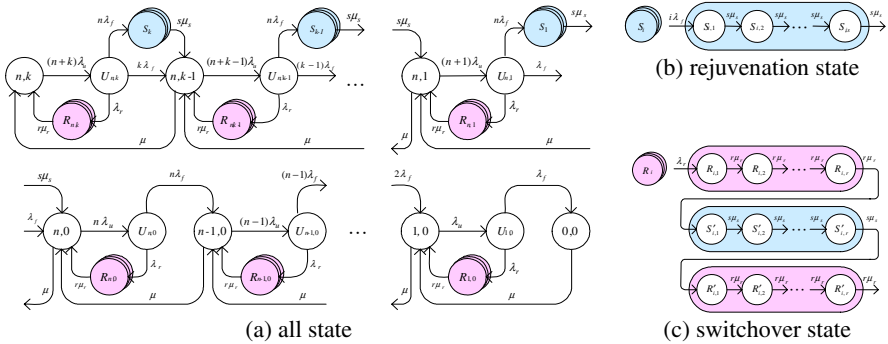


Fig. 3. State transition diagram for availability analysis of a ubiquitous computing system

The state transition diagram of the ubiquitous computing system that takes into account software rejuvenation and switchover is presented in Fig. 3. The two-tuple  $\{(n,k), (n,k-1), \dots, (1,0)\}$  annotated in each circle represents the number of primary and backup components in normal states respectively. After a certain period of mission time, the system may transit from one of the normal states to one of the unstable states  $\{U_{n,k}, U_{n,k-1}, \dots, U_{n,0}, \dots, U_{1,0}\}$  with rate  $i \cdot \lambda_u$  where  $i$  is the number of components,  $i = n + k$  ( $n$ : the number of active components,  $k$ : the number of backup components,  $\lambda_u$ : the unstable rate of altering from healthy to faint). If a component is in an unstable state, its state can change to 1) the *rejuvenation* state, at rate  $\lambda_r$  of operating the rejuvenation process, so that all components become free from faint conditions due to software aging, 2) the *switchover* state, at rate  $n \cdot \lambda_f$ , so that one backup component takes over the task of a faulty primary component when a primary component fails, or 3) the *shut-down* state at the rate  $m \cdot \lambda_f$  where a backup component is shut down ( $m$ : the number of backup components), when a backup component fails. After rejuvenation, the system state departs from one of the unstable states. In rejuvenation states  $\{R_{n,k}, R_{n,k-1}, \dots, R_{1,0}\}$ , not only one half of primary components,  $\lceil n/2 \rceil$ , but also one half of backup components,  $\lceil k/2 \rceil$ , are rejuvenated and followed by

the other half. During transitions of the software rejuvenation, it is necessary for primary and backup components to subsume for each other of their computing tasks so that computing service would not be stopped at any time. Thus, one half of primary components waiting for the rejuvenation must keep service and one half of backup components waiting for the rejuvenation take over services of one half of primary components being rejuvenated. The state (0,0) denotes that all components fail.

$$P_{n+k} = \left[ \begin{aligned} & 1 + \sum_{i=0}^{n+k-1} \left\{ \left( \frac{\lambda_f \cdot \lambda_u}{\mu} \right)^{n+k-i} \cdot \left( \prod_{l=0}^{n+k-1-i} \frac{(n+k-l)^2}{\lambda_r + (n+k-l) \cdot \lambda_f} \right) \cdot \left( 1 + \frac{\mu}{(i+1) \cdot \lambda_f} \cdot \left( 1 + \frac{2 \cdot \lambda_r}{r \cdot \mu_r} + \frac{\lambda_r}{s \cdot \mu_s} \right) \right) \right\} \\ & + \sum_{i=0}^k \left\{ \left( \frac{\lambda_f \cdot \lambda_u}{\mu} \right)^{k-i} \cdot \left( \prod_{l=0}^{k-i} \frac{(n+k-l)^2}{\lambda_r + (n+k-l) \cdot \lambda_f} \right) \cdot \frac{\lambda_r + (n+i) \cdot \lambda_f}{(n+i)^2} \cdot \frac{n \cdot \mu}{(n+1+i) \cdot \lambda_s} \right\} \\ & - \left\{ \frac{n \cdot \mu}{(n+k+1) \cdot \lambda_s} + \left( \frac{\lambda_f \cdot \lambda_u}{\mu} \right)^{n+k} \cdot \left( \prod_{l=0}^{n+k-1} \frac{(n+k-l)^2}{\lambda_r + (n+k-l) \cdot \lambda_f} \right) \cdot \frac{\mu}{\lambda_f} \cdot \left( \frac{\lambda_r}{r \cdot \mu_r} + \frac{\lambda_s}{s \cdot \mu_s} \right) \right\} \end{aligned} \right]$$

We define the *performable availability* (PA) as the probability that the systems are in one of the availability states that meet a minimum performance threshold, such as the task waiting time. The time span difference in the order of magnitude between availability and waiting time analyses is very large. To overcome the difficulty, we distinguish the definition of PA from the conventional definition of availability (CA) as below equation (1).

$$CA = \sum_{i=1}^{n-1} (P_{i,0} + P_{U_{i,0}}) + \sum_{i=0}^k (P_{n,i} + P_{U_{n,i}}) + \sum_{i=2}^{n+k} \sum_{j=1}^r (P_{R_{i,j}} + P_{R'_{i,j}}) + \sum_{i=n}^{n+k-1} \sum_{j=1}^s P_{S_{i,j}} = 1 - \left( \sum_{i=2}^{n+k} P_{S'_i} + P_{R_1} + P_0 \right) \quad (1)$$

Let  $\Delta_i$  denote the indicator to determine that the system is available when  $i$  primary components are functioning. Then,  $\Delta_i = 1$ , if  $d \leq D_u$  and  $\Delta_i = 0$ , otherwise where  $D_u$  denotes user-specified maximum waiting time, and  $d$  the waiting time when the systems have  $i$  primary components. We assume that the interarrival time of the transactions and the length of time required to process a transaction are independently and exponentially distributed random variables with rate  $1/\alpha$ ,  $1/\beta$ , respectively. The systems must have at least  $m_0$  primary components functioning in order to meet the performability requirements with respect to task waiting time. And  $\alpha/m_0\beta < 1$  should be satisfied for the condition of ergodicity. Based on these assumptions, the system performability for PA can be solved by using an M/M/ $m_0$  queueing system.  $W_q(t)$ , the waiting time distribution of M/M/ $m_0$ , is referred to as *Erlang's C formula*,  $C(m_0, \alpha/\beta)$  [14]. From this formula, the system is considered available if the condition of equation (1) is met.  $W(d) = P[W_q \leq d] \geq \psi$  is obtained, where  $\psi$  (psi) and  $W_q$  represent acceptance ratio and a random variable which describes the amount of sojourn time in queues, respectively (equation (2)). From equations (1) and (2), performable availability (PA) is defined as below:

$$PA = \begin{cases} \sum_{i=1}^{n-1} \Delta_i (P_{i,0} + P_{U_{i,0}}) + \Delta_n \sum_{i=0}^k (P_{n,i} + P_{U_{n,i}}) + \Delta_{n-1} \left( \sum_{i=2}^{n+k} \sum_{j=1}^r (P_{R_{i,j}} + P_{R'_{i,j}}) + \sum_{i=n}^{n+k-1} \sum_{j=1}^s P_{S_{i,j}} \right) & \text{if } n > m_0 \\ \Delta_n P_{n,0} & \text{if } n = m \end{cases} \quad (3)$$

Therefore, we merge the output of  $M/M/m_0$  queueing analysis with that of the steady-state probabilities of a ubiquitous computing system. At system initialization, the user sets the requirement of system availability ( $\Omega$ , omega) and derives a utilization metric ( $\rho = \alpha / m_0 \beta$ ) through periodic sampling of the average transaction arrival rate. Initially, the number of primary components satisfy the (user-defined) waiting time deadline ( $D_u$ ) through Erlang's C formula (see equation (2)). Next, the decision is made over optimal redundancy levels by analyzing the performable availability with change of the number of primary or backup components (see equation (3)). The algorithm produces the optimal redundancy level of a ubiquitous computing system meeting the performability and availability as its output. When the systems need to handle heavy workload, the dynamic configuration algorithm should follow certain priority policy in its decision making process. For example, if the priority of static services is higher than that of dynamic services, more servers would be added to the systems for static service by using equation (4), where  $S(t)$  is the number of servers for static service at time  $t$ ,  $A_s(t)$  the system availability for static service at time  $t$ ,  $L_s(t)$  the total workload in both systems,  $M_s(t-1)$ : the maximum connection number of one server, and  $S(t-1)$  the number of servers for static service at time  $t-1$ .

$$\text{if } L_s(t) > S(t) \cdot M_s(t-1) \text{ or } A_s(t) < \Omega, \text{ then } S(t) = S(t-1) + 1 \tag{4}$$

### 4 Performance Evaluation

By using the system and workload parameters reported in Table 1, we can create simulation models based on well established traffic distributions. Similar to numerous observations on the network traffic, M. E. Crovell also characterized the web traffic as bursty[15]. As a result, we use heavy-tailed distributions to characterize the high variability and self-similar nature of the traffic patterns. The time to process a static request is proportional to the files size. The service time for a dynamic object is modeled according to hyper-exponential distribution. We performed experiments to find the optimal configuration for the ubiquitous computing systems using the following system-operating parameters [8,9,10]. The systems are analyzed at intervals of 1 year. Failure rate of a component is 2 times per year and repair time 12 hours. Rejuvenation schedule is 1 time per month and healthy systems may enter unstable states every 15 days. Switchover and rejuvenation time are 10 minutes and 20 seconds, respectively. As usual, the number of transactions varies with the time of the day. We assume that the average transaction rates are 4200 transactions/hour for

**Table 1.** Averages and standard deviations of the three daily periods

Periods	Mean number of requests		Standard deviation		Average	Standard Deviation
	2005/01/22	2005/01/23	2005/01/22	2005/01/23		
0:00 – 8:00	31.50	26.75	19.71	19.40	29.13	3.36
8:00 – 16:00	135.75	137.75	42.52	44.39	136.75	1.41
16:00 – 24:00	93.37	82.00	15.88	16.23	87.63	7.95

0AM-8AM, 5400 transactions/hour for 8AM-4PM., and 4800 transactions/hour for 4PM-0AM. Each component can process 1200 transaction per hour, and at least 90% of transactions have to meet 1-minute waiting time deadline. The systems must be available up to 99.99%. To adequately approximate a deterministic sojourn time in rejuvenation and switchover states, the number of stages is set to 20.

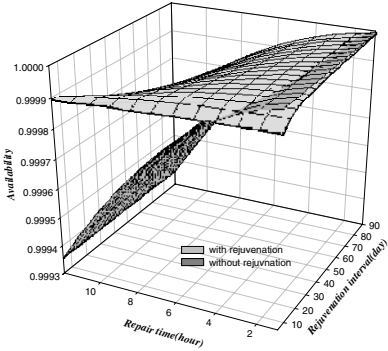


Fig. 4. System availability with respect to rejuvenation period and repair time

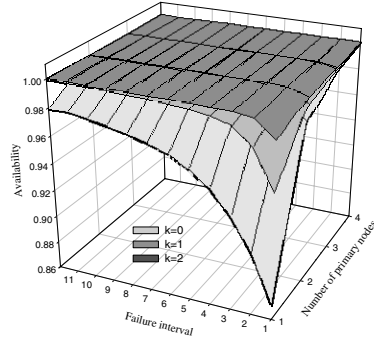


Fig. 5. Availability according to software rejuvenation process

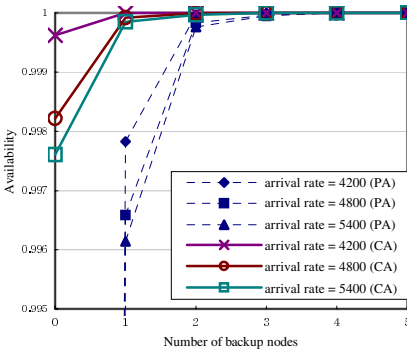


Fig. 6. Analysis according to transaction arrival rate

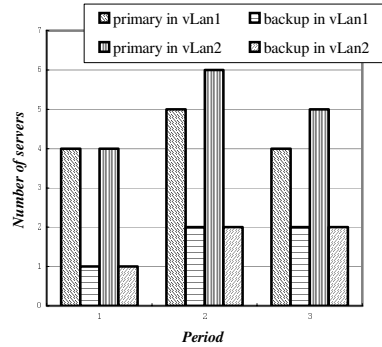


Fig. 7. The number of primary and backup server

The effect of software rejuvenation on availability of the systems is shown in Fig. 4, using the conventional availability defined in equation (1). As repair time becomes shorter, availability increases regardless of software rejuvenation, a result of software aging phenomenon, called *heisenbugs*. We note, however, systems with software rejuvenation can keep high availability because dormant software faults diminishes. In other words, software rejuvenation can reduce the frequency of failure in the ubiquitous computing systems. However, the result suggests that the system cannot meet its availability requirements of 99.99% if software rejuvenation period is too long. Fig. 5 shows the effects of failure rate and the number of backup

components on the system availability. Spare system capacity is a necessity for software rejuvenation, implying that at the single system state (1,0) software rejuvenation cannot improve availability. A duplex (1,1) or multiplex ( $v, w$ ) system with a higher degree of redundancy is needed for availability improvement, where  $v \geq 2$  and  $w \geq 2$ . However, an excessively redundant system only provides marginal improvement on availability. For instance, the availability difference between a (3,1)-way system from that of a (3,2), (4,1) or (4,2)-way system is very small. It is highly desirable to find a minimally configured system (of minimal cost) that can meet the system requirements. In Fig. 6, the *CA* and *PA* measures in three time slots are plotted with respect to transaction arrival rate, given that that the system has four primary components. The availability of a system with no backup components is too little to meet availability requirement (99.99%). However, no further visible advantage can be gained if more than four backup components are added to the system. Fig. 7 describes how to optimize a system autonomously. Initially, the algorithm selects the (7,0) to meet the waiting time deadline requirement. Next, alternatives to the initial choice, such as (6,1), (5,2) and (4,2) are examined in order to find the optimal point. After the search and computational processes, one can reduce to a (3,2)-way configuration from the (7,0)-way configuration without violating any requirements.

## 5 Conclusion

The ubiquitous computing systems that provide computing services to customers, dynamically need to meet not only the availability but also the performance requirements. In this paper, we develop a management mechanism for applying software rejuvenation, a proactive management method, into a ubiquitous computing system. It can automatically collect, aggregate, filter, track, analyze, and make plan to manage the ubiquitous computing system. Also, we validate the autonomic fault management scheme based on a workload model derived from the system log analysis with a scenario applicable to the run-time configuration phase. Moreover, this behavior such as human autonomous nervous system can be extended to other two phases of the application life cycle: the design phase and the deployment phase. In the future work, we will extend the proposed mechanism with an inference machine.

## References

1. Satyanarayanan M.: Pervasive Computing: Vision and Challenges. IEEE Personal Communications (2001) 10-17.
2. Sun\_Microsystems: J2EE Platform Specification (2002) <http://java.sun.com/j2ee/>
3. Microsoft: The Microsoft .NET Framework. Microsoft Press (2001)
4. Garg S., Moorsel A., Vaidyanathan K., Trivedi K.: A Methodology for Detection and Estimation of Software Aging. Proceedings of 9<sup>th</sup> IEEE International Symposium on Software Reliability Engineering (1998) 282-292.
5. Sullivan M., Chillarehe R.: Software Defects and Their Impact on System Availability-A Study of Field Failures in Operating Systems. Proceedings of 21<sup>st</sup> IEEE International Symposium on Fault-Tolerant Computing (1991) 2-9.

6. Scott D.: Making Smart Investments to Reduce Unplanned Downtime. Tactical Guidelines Research Note TG-07-4033, Gartner Group (1999)
7. Huang Y., Kintala C., Kolettis N., Fultion N.: Software Rejuvenation: Analysis, Module and Applications. Proceedings of 25<sup>th</sup> IEEE International Symposium on Fault-Tolerant Computing (1995) 318-390.
8. Trivedi K., Vaidyanathan K., Popstojanova K.: Modeling and Analysis of Software Aging and Rejuvenation. Proceedings of IEEE 33<sup>rd</sup> Annual Simulation Symposium (2000) 270-279.
9. Park K., Kim S.: Availability Analysis and Improvement of Active/Standby Cluster Systems using Software Rejuvenation. The Journal of Systems Software, Vol. 61, No. 2. (2002) 121-128.
10. Choi C., Kim S.: Self-configuring Algorithm for Software Fault Tolerance in (n, k)-way Cluster Systems. Lecture Notes in Computer Science, Springer, Vol. 2667, No. 1. (2003) 742-751.
11. DMTF Inc.-Common Information Model. <http://www.dmtf.org/standards/cim/>
12. Lanfranchi G., et. al.: Toward a New Landscape of Systems Management in An autonomic Computing Environment. IBM System Journal, Vol. 42, No. 1. (2003) 119-128
13. J. Arlat, et. al.: Dependability of COTS Microkernel-based Systems. IEEE Transactions on Computers, Vol. 51, No. 2. (2002) 138-163.
14. Kanoun K., Borrel M.: Fault-tolerant Systems Dependability: Explicit Modeling of Hardware and Software Component-Interactions. IEEE Transactions on Reliability, Vol. 49, No. 4. (2000) 363-376.
15. Kleinrock, L.: Queueing Systems Volume I: Theory. Wiley (1975)
16. Crovelli M. E. and Bestavros A.: Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes, IEEE/ACM Transactions on Networking, Vol. 5, No. 6. (1997) 835-846.

# Reassignment Scheme of an RFID Tag's Key for Owner Transfer

Junichiro Saito<sup>1</sup>, Kenji Imamoto<sup>1</sup>, and Kouichi Sakurai<sup>2</sup>

<sup>1</sup> Graduate School of Information Science and Electrical Engineering,  
The Department of Computer Science and Communication Engineering,  
Kyushu University,

6-10-1 Hakozaki, Higashi-ku, Fukuoka, 812-8581, Japan

saito@itslab.csce.kyushu-u.ac.jp

imamoto@itslab.csce.kyushu-u.ac.jp

<sup>2</sup> Department of Computer Science and Communication Engineering,  
Kyushu University,

6-10-1 Hakozaki, Higashi-ku, Fukuoka, 812-8581, Japan

sakurai@csce.kyushu-u.ac.jp

**Abstract.** A Radio-Frequency-Identification (RFID) tag is a small and cheap device which is combined in IC chip and an antenna for radio communications. The RFID tag is used for management of goods and used as a substitute for a bar code. However, RFID system may infringe on a consumer's privacy because it has a strong tracing ability. In this paper, we propose a key change scheme which can prevent previous owner from reading the RFID tag after changing its owner. By using our scheme, previous owner cannot read and trace ID information on the RFID tag. Moreover it is possible to combine other privacy protection scheme with our scheme because our scheme uses only symmetric key cryptography.

## 1 Introduction

A Radio-Frequency-Identification (RFID) tag is a small and cheap device that consists of an IC chip and an antenna which communicate by radio frequency. A radio communication device called reader emits a query to an RFID tag and read ID of an RFID tag. When a reader emits a query, there are readers which also transmit power for the RFID tags, so an RFID tag does not have power supply in that case. Therefore an RFID tag expected to be used as a substitute of a bar code in the future. In order to use as a bar code, the cost of an RFID tag is \$0.05/unit, and tags are small as 0.4mm \* 0.4mm and thin enough to be embedded in paper. For this reason, the capacity of a RFID tag is limited and an RFID tag is difficult to process complicated procedure. Moreover, because of using a radio frequency, a communication between an RFID tag and a reader will be simply monitored.

There are some kinds of an RFID tag and it is classified according to communication distance, the kind of memory, and the existence of a power supply. First, there are a close type (0 - several mm) and proximity type (several mm - dozens cm) and a remoteness type (dozens cm - several m) in communication distance. In

memory type, there are read only type, and once write and read type, and write and read type. In the case of writeable memory, a radio communication device called reader/writer can write in ID information on an RFID tag. In the case of read only type, the ID is determined in the manufacture stage of a RFID tag. Moreover, there are an active type and a passive type in the power supply of a RFID tag. The active type contains the power supply in the RFID tag, and a passive type obtains electric power from a reader by method which was mentioned above. From cost or the ease of use, a passive type of power supply is used in many cases.

This tag is used for next generation barcode in the field of distribution. By using an RFID tag, we can manage ID information in a database and optimize distribution and stock of products. An RFID tag has more information than printed barcode. Moreover, when an RFID tag is attached to various goods, consumers will use it after buying goods. For example, we can use an RFID tag for theft detection and goods management. More specifically, a refrigerator which reads an RFID tag can observe best-before date of foodstuffs.

However, we should concern about security problem on an RFID tag. Most important problem is privacy problem. The communication between a reader and a RFID tag is performed by radio. Thus, it is simply tapped by an attacker. Moreover, the location of the owner can be traced by tracing the information on the specific RFID tag even if the attacker cannot understand the contents of ID. This privacy about owner's location is called as location privacy. For this reason, if an RFID tag is used for distribution, leakage of information about distribution is important. Moreover, a retailer can trace a consumer after selling goods. Therefore, we need a scheme to prevent from reading ID information on an RFID tag after changing its owner.

In this paper, we propose owner change schemes for an RFID tag by using three party model and two party model. In our schemes, ID information on an RFID tag is encrypted by using symmetric cryptosystem to prevent from leakage of ID information. Moreover, our schemes can prevent from reading ID information by previous owner by changing key of encryption.

## 2 Privacy Problems on an RFID Tag

The communication between a reader and an RFID tag is performed by radio. Thus, it is simply tapped by an attacker. The reader can simply derive information from the RFID tag and it can be used to infringement of the privacy. There are two privacy problems on the RFID tag. First is the leakage of ID information. Since the RFID tag has unique ID, if the attacker obtains the ID, he can get information about objects that the tag was attached. For example, the size and the price of clothes, the contents of a wallet, the inventory information about the goods of a store etc. can be leaked. As a result, it infringes on the owner's privacy. We can protect this problem by using anonymity of ID information by using encryption scheme. Therefore, the attacker can not know what encrypted ID means.

Second problem is the tracing ID information. An attacker can trace by tracing the information on the specific tag even if an attacker cannot understand the



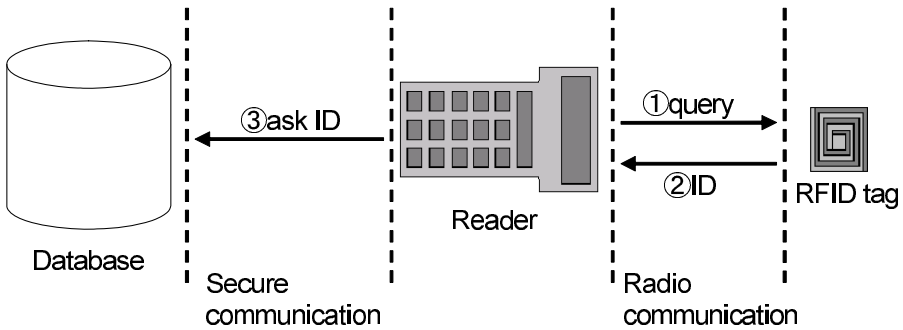


Fig. 1. RFID system

meanings of ID of the RFID tag. Therefore, he can know the location of the tag and the owner of the tag. This privacy about owner's location is called as location privacy. Therefore, an attacker can trace the owner of RFID tags by reading them. Moreover, it is also considered that the owner suffers the further damage. When a RFID tag is embedded in banknotes, it is possible that the information on an RFID tag can be read after drawing money from a bank, and it can be pursued exactly, and will be involved in a crime [1, 2]. Even if the attacker can not know what kind of banknotes a person has, he can know the person has many tags which is attached to banknotes. So RFID tags can be a detector of rich people. Therefore, the effective privacy protection scheme in an RFID tag is needed. Against these problems, some previous results [3, 4, 5] proposed privacy protection schemes which can change ID information periodically to protect location privacy. Since ID information is not fixed by using these schemes, the attacker cannot trace specific tags.

### 2.1 Privacy Problems Related with Owner Changing

When an RFID tag is used everywhere, the owner of the RFID tag is changed in its lifecycle. For example, when an RFID tag is attached to products, its owner changes from a manufacturer to a distributor, and from a retailer to a consumer. In this case, there are some problems if previous owner can read ID information on the RFID tag after changing its owner. For example, the retailer can trace the consumer. There are some schemes to prevent a third party from reading ID information [3, 4, 5]. However, since the previous owner might have important information like a decryption key, he can trace the RFID tag by using these information. Thus, we cannot protect consumer's privacy by using these schemes. Therefore, we need to change these information when owner of tags is changed.

### 2.2 Traceability and Location Privacy

We will show the difference between traceability and location privacy. RFID tags realize traceability by tracing ID information on RFID tags. When RFID tags are

used for goods management, reading ID information on RFID tags is recorded in a database and its source and transportation history are recorded. After goods are displayed at a shop, consumers can check a source of goods and know transportation history of the goods by accessing the database. In this case, this property that we can get backward information is called as traceability.

On the other hand, it is a problem that a shop assistant can trace RFID tags attached to goods by reading its ID information after consumers buy them. If the shop assistants can trace RFID tags, she can trace consumers and know the location of them. So we want that no one can get forward information after the owner of RFID tags is changed. So its privacy problem is called as location privacy.

Then, we propose schemes which can change a key on an RFID tag when its owner is changed. By using our schemes, previous owner cannot trace ID information on the RFID tag and we can protect privacy related with owner changing.

### 3 Owner Change Scheme on Three Party Model

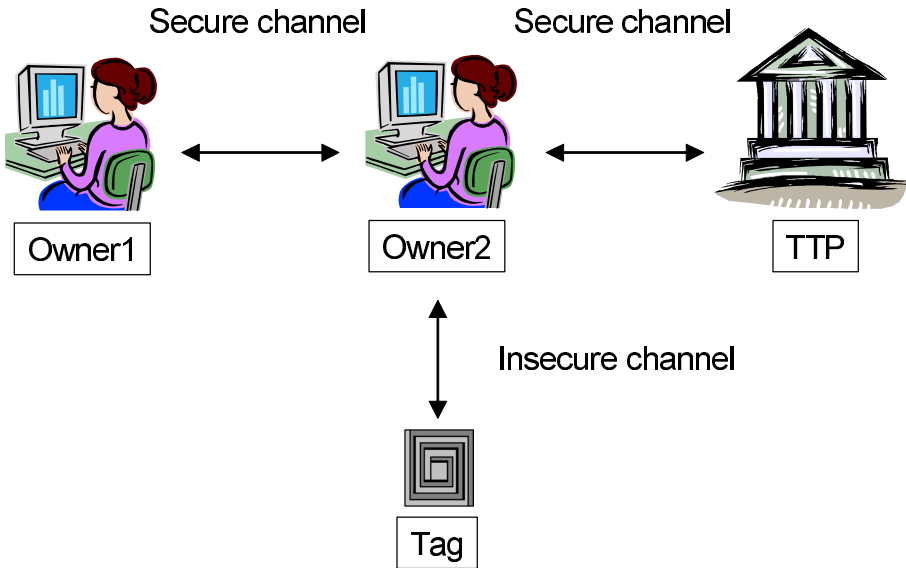
We propose an owner change scheme using symmetric key cryptosystem. In our proposed scheme, an RFID tag is required to encrypt its ID information by using symmetric key. Moreover, our scheme can be used with other privacy protection scheme like [3, 4, 5] because we propose only key change scheme. So the ID information can be a ciphertext of these privacy protection scheme. In our scheme, previous owner gives an encryption key to new owner and new owner replaces the key to a new key. As a result, previous owner cannot decrypt ID information on the RFID tag by using his own key after changing the encryption key. Moreover, when new owner changes the key, we uses trusted third party (TTP) to encrypt new key. Therefore, new owner submits the encrypted key to the RFID tag. So if an attacker eavesdrops the encrypted key, she cannot decrypt the key. Therefore, we can prevent the attacker from getting new key.

Next, we define our model in our scheme and show our scheme.

#### 3.1 Model

We show our model below. There are four entities in our model.

- $ID$  : ID is a static identifier, such that a serial number. Moreover, we can also use ciphertext of other privacy protection scheme as ID.
- $T$  : An RFID tag. It shares a symmetric key  $K_1$  with  $Owner_1$  and a symmetric key  $K_{TTP}$  with  $TTP$ . It generates a ciphertext  $C = SK_{K_1}[ID]$  by using  $K_1$ .
- $Owner_1$  : Previous owner. He shares a symmetric key  $K_1$  with the RFID tag.
- $Owner_2$  : New owner. He receives  $K_1$  from previous owner and generates new symmetric key  $K_2$ .



**Fig. 2.** Model of communication channels

- $TTP$  : Trusted third party. He shares  $K_{TTP}$  with an RFID tag  $T$ . In practice, the role is played by a service provider of traceability or ID provider.

These shared keys are securely shared in advance.

Next, we show our models of communication channels. This model is shown in Figure 2.

- $Owner_1$  to  $Owner_2$  : This channel is an existent secure channel.  $Owner_2$  can authenticate  $Owner_1$  and send a message without leakage.
- $Owner_2$  to  $TTP$  : This channel is the same as above channel.
- $Owner_2$  to  $T$  : This channel is radio frequency channel. An attacker can tap this channel. Therefore, the channel is not secure.

### 3.2 Protocol

We show our proposed scheme below.

1.  $Owner_1$  gives a key  $K_1$  to  $Owner_2$  by using a secure communication channel.
2.  $Owner_2$  generates a new key  $K_2$  and send  $K_1$  and  $K_2$  to  $TTP$  by using a secure communication channel.
3.  $TTP$  generates a ciphertext  $C_{TTP} = SK_{K_{TTP}}[K_1, K_2]$  by using a key  $K_{TTP}$  and send the ciphertext to  $Owner_2$ .
4.  $Owner_2$  send the ciphertext  $C_{TTP}$  to  $T$ .
5.  $T$  decrypts  $C_{TTP}$  by using  $K_{TTP}$ . If  $K_1$  is true,  $T$  changes the previous key  $K_1$  to the new key  $K_2$ .

The protocol is shown in Figure 3.

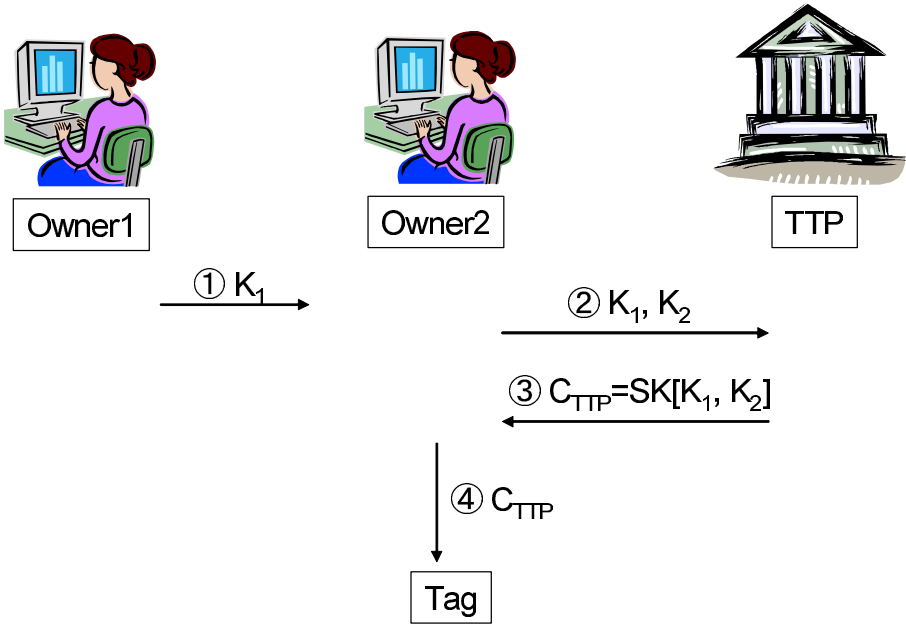


Fig. 3. Owner change scheme in three party model

### 3.3 Discussion

In our scheme, we can change the key on the RFID tag by using TTP. We suppose that the communication channels between previous owner and new owner, and between new owner and TTP are secure respectively. Therefore, an attacker can tap only the communication between new owner and the RFID tag. Moreover, in the communication between new owner and the RFID tag, the new key is encrypted by using the TTP’s key which is shared the RFID tag with TTP. Thus, previous owner cannot know the new key because the new key is encrypted by the TTP’s key.

However, new owner can know a movement history of previous owner because previous owner must give the own key to new owner in our scheme. To solve this problem, previous owner can change a key which should be sent to new owner. In our scheme, we can change a key by ourselves. Therefore, new owner cannot know the movement history of previous owner because previous owner can change a key to change an owner.

Moreover, we can prevent ID information from leaking because ID information is encrypted by symmetric key encryption. About tracing ID information, we can prevent by changing the key frequently. Since the ciphertext of ID information is changed by changing the key, we can protect location privacy. However, our scheme is not suitable to changing the key frequently because we must communicate with TTP. But, we can protect location privacy by combining our scheme with other privacy protection scheme like [3, 4, 5].

Another problem is tampering with the RFID tag. Since the RFID tag has the symmetric key, an attacker can get the key by tampering with the RFID tag. To protect from tampering, we can use tamper-resistant device to keep the key. However, tamper-resistant device is too expensive for the RFID tag to be used for goods management.

## 4 Owner Change Scheme on Two Party Model

Next, we propose an owner change scheme on two party model. This scheme can be also used with other privacy protection scheme like [3, 4, 5]. In this scheme, we can change a key of an RFID tag without TTP.

### 4.1 Model

We show our model below.

- $T$  : An RFID tag. It has ID information and a key  $K_1$  which is shared with  $Owner_1$ . It generates a ciphertext  $C = SK_{K_1}[ID]$  using  $K_1$ . Moreover, it generates a nonce  $N$  by using a random number.
- $Owner_1$  : Previous owner. It shares a key  $K_1$  with  $T$ .
- $Owner_2$  : New owner. It receives  $K_1$  from  $Owner_1$  and generates a new key  $K_2$ .

These shared keys are securely shared in advance.

Moreover, we suppose a forward channel and a backward channel [6]. The forward channel is a communication channel from a reader to an RFID tag. It

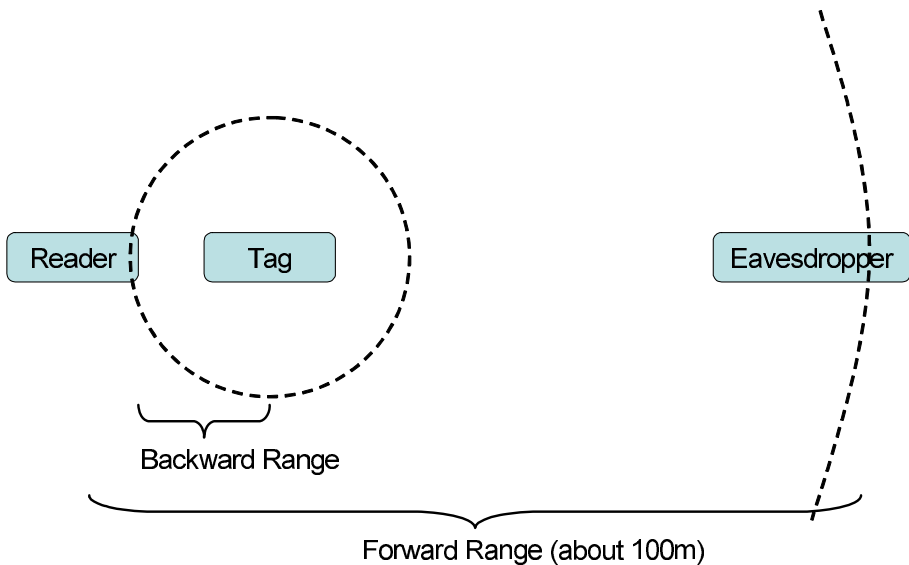


Fig. 4. Forward channel and backward channel

is relatively long range. On the other hand, the backward channel is a communication channel from an RFID tag to a reader. It depends on the capability of the RFID tag. However, the backward channel is shorter than the forward channel. Therefore, it is more difficult to tap the backward channel than the forward channel. We show the relationship between the forward channel and the backward channel in Figure 4.

Next, we show our models of communication channels.

- $Owner_1$  to  $Owner_2$  : This channel is an existent secure channel.  $Owner_2$  can authenticate  $Owner_1$  and send a message without leakage.
- $Owner_2$  to  $T$  : This channel is a forward channel by using radio frequency. It is possible to be tapped by an attacker. So, this channel is not secure.
- $T$  to  $Owner_2$  : This channel is a backward channel by using radio frequency. It is possible to be tapped by an attacker. However, the attacker must approach to tap this channel because this channel is backward channel. Therefore, this channel is more secure than the communication channel from  $Owner_2$  to  $T$ .

### 4.2 Protocol

We show our proposed scheme below.

1.  $Owner_1$  gives a key  $K_1$  to  $Owner_2$  by using a secure communication channel.
2.  $Owner_2$  sends a query to  $T$ .
3.  $T$  generates a nonce  $N$  and sends it to  $Owner_2$ .

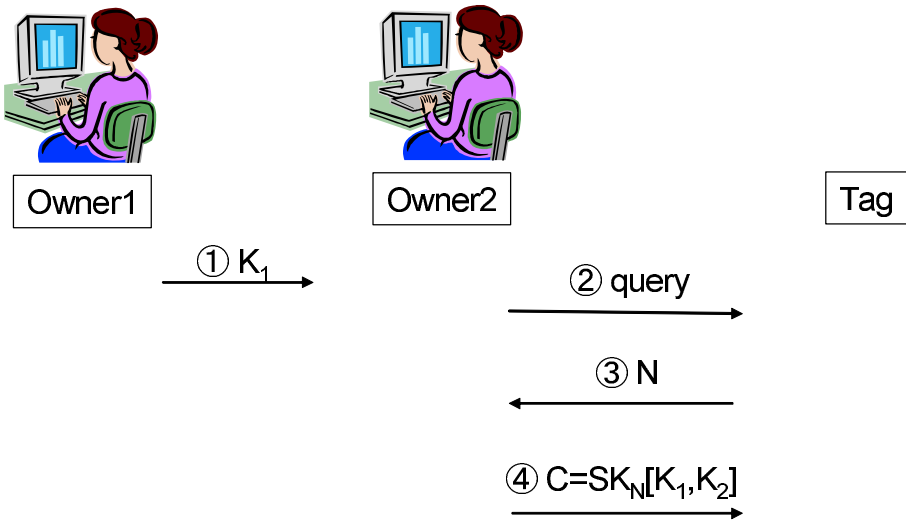


Fig. 5. Owner change scheme in two party model

4.  $Owner_2$  generates a new key  $K_2$  and generates a ciphertext  $C = SK_N[K_1, K_2]$ .  $Owner_2$  send the ciphertext  $C$  to  $T$ .
5.  $T$  decrypts  $C$  by using  $N$ . If  $K_1$  is true,  $T$  changes a key  $K_1$  to  $K_2$ .

The protocol is shown in Figure 5.

### 4.3 Discussion

In our scheme, new owner receives a symmetric key from previous owner and changes the key on an RFID tag by using the key. Moreover, we can protect the new key by using a nonce as an encryption key. When the RFID tag sends the nonce, it is difficult to tap the nonce because the communication is backward channel. However, since backward channel is radio frequency, an attacker can approach and tap the nonce. Therefore, when the RFID tag sends the nonce, we can use more short range communication channel like contact communication.

Moreover, since we can change a key by ourselves, previous owner can change a key to send it to new owner.

## 5 Application

We show application for our schemes. Our schemes are fit to goods management. When RFID tags are used for goods management, its owner changes from a manufacturer to a distributor, and from a retailer to a consumer. A database keeps history of goods movements. So when consumer buy goods, she can get information about a source of goods and operations in manufacture by accessing the database. But she want that the retailer cannot know RFID tags' movement any more. By using our schemes, she can change keys of RFID tags and prevent the retailer from reading RFID tags.

## 6 Conclusion

In this paper, we proposed owner change schemes to protect a new owner's privacy. We proposed two schemes, first is an owner change scheme on three party model and second is an owner change scheme on two party model. In the scheme using TTP, we can prevent a new key from leaking by using a key which is shared an RFID tag with TTP. In the second scheme, we can prevent a key from leaking by supposing backward channel.

As a future work, we will evaluate the cost of implementation of our proposed schemes.

## References

1. A. Juels and R. Pappu, "Squealing Euros: Privacy Protection in RFID-Enabled Banknotes," In R. Wright, editor, Financial Cryptography '03, Springer-Verlag, 2003.
2. Gildas Avoine, "Privacy Issues in RFID Banknote Protection Schemes," International Conference on Smart Card Research and Advanced Applications - CARDIS, Toulouse, 2004.

3. Junichiro Saito, Jae-Cheol Ryou and Kouichi Sakurai, "Enhancing privacy of Universal Re-encryption scheme for RFID tags," EUC2004, LNCS Vol.3207.
4. A. Juels, "Minimalist Cryptography for RFID tags," Fourth Conference on Security in Communication Networks(SCN'04), 2004.
5. Miyako Ohkubo, Koutarou Suzuki, Shingo Kinoshita, "Cryptographic Approach to a Privacy Friendly Tag," RFID Privacy Workshop, 2003.
6. S. A. Weis, S. Sarma, R. Rivest, and D. Engels, "Security and privacy aspects of low-cost radio frequency identification systems," In First International Conference on Security in Pervasive Computing, 2003.
7. A. Juels, R. L. Rivest and M. Szydlo, "The Blocker Tag: Selective Blocking of RFID tags for Consumer Privacy," ACM Press, 2003.



# Author Index

- Abu-Ghazaleh, Nael B. 785  
Aghajan, Hamid 1069  
Ahn, Sungjin 1148  
Albayrak, Sahin 756  
Amamiya, Makoto 433  
Angaman, Eric 592  
Angelides, Marios C. 556  
Anusha, S. 1047  
Anzai, Jun 894  
Asaka, Takuya 1201  
Audsley, Neil C. 632
- Bacon, Jean 652  
Baek, Joo-Young 734  
Baek, Yunju 622  
Baig, Z.A. 208  
Baik, Doo-Kwon 161  
Baquer, M. 208  
Barolli, Leonard 238, 443  
Bhalla, Subhash 509  
Birari, Shailesh M. 1036  
Bodhuin, Thierry 21  
Bunder, Martin 423
- Callaghan, Vic 345  
Canfora, Gerardo 21  
Capi, Genci 538  
Cha, Hojung 1191  
Chang, Han-Bin 547  
Chang, Hsuan-Pu 171  
Chang, Tsung-Hsiang 393  
Chang, Wen-Chih 171  
Chatzikokolakis, Konstantinos 744  
Chen, HuaJun 326  
Chen, Xiaofeng 480, 825  
Chen, Yongqiang 817  
Chen, Yu 51  
Cheng, Jingde 874  
Cheng, Shao-you 499  
Chiang, Tzu-Chiang 714  
Chin, Jeannette 345  
Chin, Su-Li 180  
Chiu, Kenneth 785  
Cho, Hyuntae 622
- Cho, Jinsung 1108  
Cho, Kyu-Hyung 976  
Cho, Yongyun 131  
Choi, Changyeol 1293  
Choi, Eun Young 945  
Choi, Jaeyoung 131  
Choi, Ji Young 470  
Choi, Kun Myon 683  
Choi, Kwangseok 1128  
Choi, Lynn 1128  
Choi, Seokhoon 693  
Choi, Soonyong 248  
Choi, Sukwon 1191  
Choi, Sung Jin 1088  
Choi, Won-Ho 91  
Choo, Hyunseung 683  
Chothia, Tom 744  
Chu, Hao-hua 499  
Chuang, Chih-Chieh 298, 452  
Chung, Kyo Il 864  
Courbot, Alexandre 81
- De Marco, Giuseppe 443  
Deolalikar, Vinay 1025  
Ding, Yan Annie 663  
Dohi, Tadashi 1283  
Doi, Yusuke 375  
Domaszewicz, Jaroslaw 642  
Durrezi, Arjan 238  
Durrezi, Mimoza 238
- Endo, Takumi 874  
Eom, Doo-seop 151  
Everitt, David 663
- Feng, Dan 403, 776  
Fountain, Tony 785  
Fujinami, Kaori 335
- Garrity, Grainne 1235  
Ghim, Soo-Joong 576  
Goi, Bok-Min 936  
González-Tablas, A.I. 797  
Goto, Yuichi 874  
Gracanic, Denis 1245

- Grimaud, Gilles 81  
 Gu, Hongliang 51  
 Gu, Yu 1178  
  
 Haga, Hirohide 306  
 Han, Dong-Guk 864  
 Han, Joohyun 131  
 Han, Sunyoung 288  
 Hanna, Edward 1235  
 Hasegawa, Masaki 509  
 Hinchey, Michael G. 1245  
 Ho, Chien-Ju 393  
 Hong, Sungjune 288  
 Hsu, David Chawei 393  
 Hsu, Han-Jen 101  
 Hsu, Hui-Huang 547  
 Hsu, Jane 393  
 Hsu, Jane Yung-jen 499  
 Hu, Hanping 817  
 Hu, Yiming 766  
 Huang, Chun-Hong 171, 180  
 Huang, Jianyong 423  
 Huang, Runhe 365  
 Huang, Xinli 724  
 Huang, Xinyi 480, 835  
 Huang, Yueh-Min 714  
 Huh, Eui-Nam 683  
 Hung, Jason C. 180  
 Hur, Kyeong 151, 470  
 Hwang, Jun 228  
 Hwang, Kwang-il 151  
  
 Ibrahim, Ismail Khalil 355  
 Imamoto, Kenji 1303  
 In, Hoh Peter 161  
 Inoue, Atsushi 375  
 Ishiyama, Masahiro 375  
 Iwamoto, Kazuki 1283  
 Iyer, Sridhar 1036, 1047  
  
 Jameel, Hassan 1225  
 Jarnjak, Fran 161  
 Jeong, Dongwon 161  
 Jia, Zhiping 845  
 Jin, Min-Sik 91  
 Jin, Qun 171  
 Jin, Wang 1108  
 Jing, Yixin 161  
 Jo, Yong-hyun 1118  
  
 Jung, Min-Soo 91  
 Jung, Wonwoo 1191  
  
 Kaio, Naoto 1283  
 Kalim, Umar 1225  
 Kaneda, Shigeo 306  
 Kang, Chul-Hee 693  
 Kang, Chung Gu 470  
 Kang, Jaewoo 198  
 Kang, Jeonil 383  
 Kawamura, Takahiro 71  
 Ke, Chia-nan 499  
 Khan, A.I. 208  
 Kim, Bum Han 926  
 Kim, Dae-Won 228  
 Kim, Daeyoung 1148  
 Kim, Dong-kyoo 1006  
 Kim, Euijik 693  
 Kim, Eunkyo 460  
 Kim, Hee Jung 1078  
 Kim, Ho Won 673, 864  
 Kim, InSu 704  
 Kim, Jin-A 734  
 Kim, Jinhyung 161  
 Kim, Joongheon 460  
 Kim, Joonmo 460  
 Kim, Jungsun 1128  
 Kim, Jungtae 986  
 Kim, Kyung Tae 1098  
 Kim, Meejoung 693  
 Kim, Moon Hae 278  
 Kim, Seok-hwan 151  
 Kim, Seungjoo 965  
 Kim, Sooyeon 460  
 Kim, Sungjin 460  
 Kim, Sungsoo 1293  
 Kim, Tae Hyun 864  
 Kim, Wonil 1006  
 Kim, Woo-Jae 734  
 Kim, Yong Suk 470  
 Kim, Yun-Sung 734  
 Kishino, Yasue 121  
 Ko, Il Seok 1265  
 Ko, Young-Bae 1138  
 Koizumi, Hisao 268  
 Koo, Jae Hyung 926  
 Kotsis, Gabriele 355  
 Krishnaswamy, Shonali 61  
 Kristiansen, Lill 316  
 Kronsteiner, Reinhard 355

- Kshemkalyani, Ajay D. 807  
 Kunito, Goro 1  
 Kurkovsky, Anatoly 141  
 Kurkovsky, Stan 141  
 Kursawe, K. 797  
 Kwon, Eui Hyeok 1169  
 Kwon, Taekyoung 996
- Lai, Elaine 1058  
 Lai, Yu-Sheng 298  
 Lanka, Rodrigo 413  
 Lee, Byungil 673  
 Lee, Cheolho 976, 986  
 Lee, Choonhwa 460  
 Lee, Dong Hoon 926, 945  
 Lee, Gunhee 1006  
 Lee, Hyang-tack 1118  
 Lee, HyungHyo 490, 704  
 Lee, Jae Yeol 258  
 Lee, Kwangwoo 965  
 Lee, Sang-Ho 884  
 Lee, Sangsoo 1148  
 Lee, Seungyong 490, 704  
 Lee, Soo Jin 278  
 Lee, Su Mi 945  
 Lee, Sun-Don 734  
 Lee, Sung-Hee 1138  
 Lee, Sung Young 566, 1108, 1225  
 Lee, Won-hee 151  
 Lee, Wonjun 460  
 Lee, Yong Hwan 1078  
 Lee, Young-Koo 566,  
 1108, 1225  
 Lee, YoungLok 490, 704  
 Lee, Yuan-Hsiang 393  
 Lee, Yun Kyung 1169  
 Lee, Yunho 965  
 Lei, Shu 1108  
 Leu, Fang-Yie 1255  
 Li, Jian Raymond 1016  
 Li, Jingyu 11  
 Li, Jyun-Sian 101  
 Li, Yin 724  
 Li, Zhao 1157  
 Liao, Yi-Chun 547  
 Lim, Jae Sung 1169  
 Lin, Chi-yau 499  
 Lin, Nigel H. 171  
 Liu, HengChang 1157, 1178  
 Liu, Zhaoyu 1016
- Loke, Seng Wai 61  
 Loreto, S. 443
- Ma, Fanyuan 724  
 Ma, Jianhua 365  
 Macindoe, Owen 189  
 Maher, Mary Lou 189  
 Mao, Hung-Jen 452  
 Marreiros, Goretí 41  
 Matsumoto, Tsutomu 894  
 Mesarina, Malena 1025  
 Mishima, Yuichiro 268  
 Miura, Junichi 874  
 Mizuno, Tadanori 111  
 Moon, Jongsub 976, 986  
 Morimoto, Shoichi 874  
 Morino, Hiroaki 905  
 Mu, Yi 480, 825, 835, 854, 1273
- Na, Yun-Ji 1265  
 Nakajima, Tatsuo 335  
 Nakamura, Hiroyuki 612  
 Nakanishi, Kei 365  
 Nanashima, Koichi 874  
 Neovius, Mats 602  
 Nevs, José 41  
 Nguyen, Hoang Nam 905  
 Nishio, Shojiro 121  
 Noh, BongNam 490, 704  
 Nyang, DaeHun 383
- Oda, Kentaro 413  
 O'Hagan, Patricia 1235  
 Oh, Y.C. 1078  
 Ohsuga, Akihiko 71  
 Okuda, Shinya 306  
 Osland, Per-Oddvar 316  
 Østhus, Egil C. 316  
 Ozaki, Satoshi 375
- Park, Byung Joon 1128  
 Park, Cheol-Min 228  
 Park, HeeMan 490  
 Park, JuSung 383  
 Park, Noseong 1148  
 Park, So-Young 884  
 Paruchuri, Vamsi 238  
 Phan, Raphael C.-W. 936  
 Pradhan, Salil 1025  
 Preziosi, Rosa 21

- Qi, Zhengwei 218  
 Qiao, Xinxiao 845  
 Qin, Lingjun 776  
 Qu, Yugui 1157, 1178  
  
 Ra, Ilkyeun 576  
 Rakotonirainy, Andry 61  
 Ramos, B. 797  
 Ramos, Carlos 41  
 Rash, James L. 1245  
 Recker, John 1025  
 Redfern, Andrew 1058  
 Ribagorda, A. 797  
 Roh, Byeong-hee 1078, 1118  
 Rój, Michał 642  
 Rouff, Christopher A. 1245  
 Ruan, Chun 1273  
 Ryu, Keun Ho 198  
  
 Sagara, Ryohei 121  
 Saito, Junichiro 1303  
 Sajjad, Ali 1225  
 Sakamoto, Kenji 1  
 Sakamoto, Kouichi 529  
 Sakurai, Kouichi 433, 916, 1303  
 Salim, Flora Dilys 61  
 Satoh, Ichiro 31  
 Sawamoto, Jun 268  
 Schizas, Christos N. 556  
 Schloter, Philipp 1069  
 Seberry, Jennifer 423, 1273  
 Seo, Dong Woo 258  
 Seo, Jungtaek 976, 986  
 Seo, Sungbo 198  
 Setozaki, Makoto 365  
 Sharma, Peeyush 1016  
 Shi, Dongyu 218  
 Shi, Yuanchun 11, 51  
 Shih, Timothy K. 171, 547  
 Shin, Dongil 248  
 Shin, Dongkyoo 248  
 Shin, Sooyeon 996  
 Shin, Sujeong 1191  
 Shon, Taeshik 976, 986  
 Siddiqi, M.U. 936  
 Sie, Yun-Long 171  
 Simplot-Ryl, David 81  
 Sofokleous, Anastasis A. 556  
 Son, Minwoo 248  
 Song, Kwanho 288  
  
 Song, Young-Mi 1138  
 Sterritt, Roy 1235, 1245  
 Suh, Young-Joo 734  
 Sunaga, Hiroshi 612  
 Susilo, Willy 423, 480, 825, 835, 854  
  
 Tabata, Toshihiro 916  
 Takagi, Tsuyoshi 864  
 Takahashi, Ken'ichi 433  
 Takahashi, Tatsuo 111  
 Takahashi, Tatsuro 1201  
 Takahashi, Toshiya 268  
 Takenouchi, Takao 71  
 Tanaka, Satoshi 1, 111  
 Tang, Chia-Tong 180, 547  
 Terada, Tsutomu 121  
 Tilak, Sameer 785  
 Tilwaldi, Dilmurat 268  
 Tortorella, Maria 21  
 Toyofuku, Tatsuya 916  
 Truong, Binh An 566  
 Truszkowski, Walt 1245  
 Tsai, Chih-Hsiao 452  
 Tsai, Ming-Hui 714  
 Tsai, Min-Shieh 393  
 Tsukamoto, Masahiko 121  
  
 Usher, Paul S. 632  
  
 Vandewalle, Jean-Jacques 81  
  
 Wakayama, Shirou 375  
 Wang, Fang 403  
 Wang, Honghao 766  
 Wang, Jhing-Fa 101  
 Wang, Mu-Chun 393  
 Wang, Wen-Yang 298  
 Wang, Ying-Hong 298, 452  
 Wepiwé, Giscard 756  
 Won, Dongho 965  
 Wright, Paul 1058  
 Wu, Jing 1273  
 Wu, Zhaohui 326  
  
 Xu, Guangyou 51  
 Xu, Shidi 854  
  
 Yamada, Naoharu 1  
 Yamane, Satoshi 1211  
 Yamazaki, Kenichi 1, 111

Yang, Hyungkyu 965  
Yang, Wei-Jie 1255  
Yang, Yoon-Sim 91  
Ye, Zhiyong 326  
Yokohata, Yuki 612  
Yokota, Masao 538  
Yoneki, Eiko 652  
Yoo, Chae-Woo 131  
Yoo, Kee-Young 586, 955  
Yoo, S.W. 1078, 1118  
Yoon, Eun-Jun 586, 955  
Yoon, Yong-Ik 576  
Yoshida, Takaichi 413  
Yoshihisa, Tomoki 121  
You, Jinyuan 218

Youm, Sungkwan 693  
Youn, Hee Yong 1088, 1098  
Zamudio, Victor 345  
Zanev, Vladimir 141  
Zeng, Lingfang 403  
Zhang, Fangguo 480, 825  
Zhang, Futai 835  
Zhang, Shunda 403  
Zhang, Wei 1157, 1178  
Zhang, Wenju 724  
Zhao, Baohua 1157, 1178  
Zhao, Qiangfu 519, 529  
Zhong, Yonil 1108  
Zhou, Qi 1201

## **Erratum to: Embedded and Ubiquitous Computing – EUC 2005 Workshops**

Tomoya Enokido, Lu Yan, Bin Xiao, Daeyoung Kim, Yuanshun Dai  
Laurence T. Yang (Eds.)

Erratum to:

Autonomic Agents for Survival Security Systems

Roy Sterritt, Grainne Garrity, Edward Hanna, Patricia O’Hagan,

DOI 10.1007/11596042\_125

“The name and email address of the second author of the paper starting on page 1235 of this volume have been removed at her own request.”

---

The online version of the book can be found at:

<http://dx.doi.org/10.1007/11596042>

---

T. Enokido, L. Yan, B. Xiao, D. Kim, Y. Dai and L.T. Yang (eds.), *Embedded and Ubiquitous Computing – EUC 2005 Workshops* DOI 10.1007/11596042\_133,

© 2005 IFIP International Federation for Information Processing

1318