Hongke Zhang
Stephan Olariu
Jiannong Cao
David B. Johnson (Eds.)

LNCS 4864

Mobile Ad-hoc and Sensor Networks

Third International Conference, MSN 2007
Beijing, China, December 2007
Proceedings

Springer

Lecture Notes in Computer Science 4864

Hongke Zhang Stephan Olariu
Jiannong Cao David B. Johnson (Eds.)

Mobile Ad-hoc and Sensor Networks

Third International Conference, MSN 2007
Beijing, China, December 12-14, 2007
Proceedings

Springer

Volume Editors

Hongke Zhang
Beijing Jiaotong University
School of Electronics and Information Engineering
Next Generation Internet Research Center (NGIRC)
Haidian, Beijing 100044, China
E-mail: hkzhang@center.njtu.edu.cn

Stephan Olariu
Old Dominion University
Department of Computer Science
Norfolk, VA 23529-0162, USA
E-mail: olariu@cs.odu.edu

Jiannong Cao
The Hong Kong Polytechnic University
Department of Computing
Hung Hom, Kowloon, Hong Kong, China
E-mail: csjcao@comp.polyu.edu.hk

David B. Johnson
Rice University
Department of Computer Science
Houston, TX 77005-1892, USA
E-mail: dbj@cs.rice.edu

Library of Congress Control Number: 2007939974

CR Subject Classification (1998): E.3, C.2, F.2, H.4, D.4.6, K.6.5

LNCS Sublibrary: SL 5 – Computer Communication Networks and
Telecommunications

ISSN 0302-9743
ISBN-10 3-540-77023-2 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-77023-7 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12197491 06/3180 5 4 3 2 1 0

Preface

The principal theme of MSN conferences is the development and deployment of protocols, algorithms, systems, and applications for mobile ad-hoc and wireless sensor networks.

Following the success of MSN 2005 and MSN 2006, MSN 2007 provided a forum for researchers and practitioners working in related areas to exchange research results and share development experiences.

MSN 2007 attracted 304 submissions. Each paper was reviewed by at least three members of the Program Committee (PC) and reviewers. The final program included 75 papers, which covered a range of different topics, including routing, network protocols, energy management, security, etc.

The Organizing Committee would like to thank the Steering Committee members Xiaohua Jia, Sajal K. Das, Ivan Stojmenovic, and Jie Wu for their support and guidance in the conference organization. We would like to take this opportunity to thank all the authors for their submissions to the conference. Many of them traveled great distances to participate in this symposium and make their valuable contributions. Thanks to all the Program Committee members for their valuable time and effort in reviewing the papers. Without their help and advice this program would not be possible. Special thanks go to the conference PC Vice-Chairs, Eric Fleury, Vojislav B. Misic, and Pedro M. Ruiz, for their hard work in assembling the international PC and coordinating the review process.

We appreciate the support from the invited speakers, Sajal K. Das, and Zhisheng Niu. Their keynote speeches greatly benefited the audience. Last but not the least, we would like to thank the Local Organization Committee Chair, Deyun Gao, Yajuan Qin and all members for making the arrangements and for organizing an attractive social program.

December 2007

Hongke Zhang
Stephan Olariu
Jiannong Cao
Dave Johnson

Organization

MSN 2007 was organized by the Next Generation Internet Research Center, School of Electronics and Information Engineering, Beijing Jiaotong University.

Executive Committee

General Co-chairs	Jiannong Cao (Hong Kong Polytechnic University, Hong Kong)
	Dave Johnson (Rice University, USA)
Program Co-chairs	Hongke Zhang (Beijing Jiaotong University, China)
	Stephan Olariu (Old Dominion University, USA)
Program Vice Chairs	Eric Fleury (CITI/ARES, France)
	Vojislav B. Misic (University of Manitoba, Canada)
	Pedro M. Ruiz (University of Murcia, Spain)
Publicity Co-chairs	Jong Hyuk Park (Hanwha, Korea)
	Paolo Bellavista (University of Bologna, Italy)
Publication Co-chairs	Deyun Gao (Beijing Jiaotong University, China)
	Zhang Ling (Tsinghua University, China)
Local Organization Co-chairs	Deyun Gao (Beijing Jiaotong University, China)
	Yajuan Qin (Beijing Jiaotong University, China)
Awards Co-chairs	Sagar Naik (University of Waterloo, Canada)
	Christian Lavault (University Paris XIII, France)
	Cristina Pinotti (University of Perugia, Italy)
Steering Committee	Xiaohua Jia (City University of Hong Kong, HK)
	Sajal K. Das (University of Texas at Arlington, USA)
	Ivan Stojmenovic (University of Ottawa, Canada)
	Jie Wu (Florida Atalantic University, USA)
Web Master	Hongwei Huo (Beijing Jiaotong University, China)

Program Committee

Marcello Dias de Amorim	University P.M. Curie, France
Nuno Preguica	Universidade Nova de Lisboa, Portugal
Antoine Fraboulet	CITI, France
Jiming Chen	Zhejing University, China
Guillaume Chelius	INRIA, France
Dongfeng Yuan	Shandong University, China
Pietro Michiardi	EURECOM, France
Vasughi Sundramoorthy	Lancaster University, UK
Gaogang Xie	ICT AC, China
Lim Teck Meng	Nanyang Tech. University, Singapore
Artur Ziviani	LNCC, Brazil
Marin Bertier	France
Pilu Crescenzi	University of Florence, Italy
Prudence W.H. Wong	University of Liverpool, UK
Andrea Passarella	CNR, Italy
Yu Chen	Texas A&M University, USA
Kui Wu	University of Victoria, Canada
Sidong Zhang	Beijing Jiasotong University, China
Samuel Pierre	Polytechnique Montreal, Canada
Ayalvadi Ganesh	Microsoft, UK
Bartek Blaszczyszyn	ENS, France
Dominique Bartel	France Telecom R&D, France
Qingfeng Huang	PARC Inc., USA
Vania Conan	THALES, France
Pingzhi Fan	SWJTU, China
Yanghee Choi	Seoul National University, Korea
Massimo Franceschetti	University of California, San Diego, USA
Carla Fabiana-Chiasserini	Dipartimento di Elettronica at Politecnico di Torino, Italy
Pietro Manzoni	Universidad Politecnica de Valencia, Spain
Sotiris Nikoletseas	CTI/University of Patras, Greece
Stefan Weber	Trinity College Dublin, Ireland
Raffaele Bruno	National Research Council, Italy
Stephane Grumbach	LIAMA, the Sino-French IT Lab, France
J.J. Garcia-Luna-Aceves	University of California Santa Barbara, USA
Cruz Vasilis Friderikos	King's College, London, UK
Miguel A. Labrador	University of South Florida, USA
Luis Munoz	University of Cantabria, Spain
David Simplot-Ryl	University of Lille, France
Ivan Stojmenovic	University of Ottawa, Canada
W.G Wu	HongKong Polytech University, Hong Kong
Violet R. Syrotiuk	Arizona State University, USA
Olivier Marce	Alcatel Lucent, France

Suprakash Datta	York University, Canada
Hongyi Wu	CACS. University of Louisiana, USA
Juan A. Sanchez	University of Murcia, Spain
Lin Cai	University of Victoria, Canada
Ling-Jyh Chen	Academia Sinica, ROC
Aysegul Cuhadar	Carleton University, Canada
Y. Charlie Hu	Purdue University, USA
Bengi Karacali	Avaya Labs, USA
Ibrahim Korpeoglu	Bilkent University, Turkey
Deepa Kundur	Texas A and M University, USA
Cheng Li	Memorial University of Newfoundland, Canada
Sahra Sedigh	University of Missouri-Rolla, USA
Weisong Shi	Wayne State University, USA
Yu Wang	University of North Carolina at Charlotte, USA
Vincent Wong	University of British Columbia, Canada
Wang Lei Rui	Washington State University, USA
Maode Ma	Nanyang Tech. University, Singapore
Xiaojiang (James) Du	North Dakota State University, USA
Natalia Stakhanova	Iowa State University, USA

Additional Reviewers

Amiya Nayak
Chao Wang
Chunhe Yu
Costas Constantinou
Deke Guo
Desheng Zhu
Deyun Gao
Dong Liu
Dongliang Xie
Du Xu
Fengyuan Ren
GuanDing Yu
Guiling Sun
Guoming Zuo
Haigang Gong
Haiyong Luo
Han-Chieh Chao
Henry Huo
Hong Tang
Hongbin Luo
Hongfang Yu
Hongyan Li
Huadong Ma
Huiyao An
Jian Yuan
Jiandong Li
Jiang Hao
Jiang Yu
Jun Liu
Justin Lipman
Kang Qi
Li Cui
Limin Fan
L. Cai
L. Zhang
Michel R.
Mikhail Nes.
Ming Liu
Naian Liu
Ngugi A.
Ningning Lu
Petar P.
Qing-An Zeng
R. Wang
Sheng Min
Shiduan Cheng
Shihong Zou
Shizhong Xu
Sh. Gao
Timothy K.
Shih Vincent
Violet R.
Wangdong Qi
W. Mansoor
W.G. Wu
Weiji Su
Weixin Xie
Wenming Cao
Xiaoping Xue
Celia X.
Yajuan Qin
Yan Ren
Yanghee Choi
Yaping Lin
Y. Sun
Ying Liu

Yingchi Mao
Yiqun Qian
Yong Tang
Yongjun Xu
Y.K Ji
Y. Li
Zenghua Zhao
Zhipeng Chang

Sponsoring Institutions

Beijing Jiaotong University
Nokia
Springer
Crossbow Technology

Table of Contents

Keynote Speech

Routing

Protocol

Energy Efficiency

Data Processing

Self-organization and Synchronization

Deployment and Application

Security

Information Intensive Wireless Sensor Networks: Challenges and Solutions*

Sajal K. Das

Center for Research in Wireless Mobility and Networking (CReWMaN),
The University of Texas at Arlington, Arlington, TX 76019-0015, USA
das@uta.edu

Abstract. Tremendous advances in embedded systems, sensors and wireless communications technology have made it possible to build large-scale wireless sensor networks (WSNs). Due to their potential applications,WSNs have attracted significant attention in the industry, academic, and government organizations. In particular, by commanding a large number of distributed and coordinated sensor nodes, WSNs can effectively act as the human-physical world interface in future digital world through sensing and actuating. However, the inherent characteristics of WSNs typified by extremely scarce resources (e.g., bandwidth, CPU, memory and battery power), high degree of uncertainty, and lack of centralized control pose significant challenges in providing the desired quality, information assurance, reliability, and security. This is particularly important for mission critical applications that involve information intensive WSNs including video sensors. In this talk, we will examine the uncertainty-driven unique challenges and key research problems in information intensive wireless sensor networks in the areas of aggregation, clustering, routing, data dissemination, coverage and connectivity, and security. We will present our novel solutions to some of these problems and conclude with future directions.

Biography

Dr. Sajal K. Das received B.S. degree from Calcutta University (1983), M.S. degree from Indian Institute of Science at Bangalore (1984), and PhD degree from University of Central Florida (1988), all in computer science. Currently he is a University Distinguished Scholar Professor of Computer Science and Engineering and the Founding Director of the Center for Research in Wireless Mobility and Networking (CReWMaN) at the University of Texas at Arlington (UTA). Dr. Das is also a Visiting Professor at the Indian Institute of Technology (IIT), Kanpur and IIT Guwahati; Concurrent Professor of Fudan University in Shanghai and Advisory Professor of Beijing Jiaotong University, China; and Visiting Scientist at the Institute of Infocomm Research (I2R) in Singapore.

* This work is supported by NSF grants IIS-0326505, CNS-0421282 and CNS-0721951, and Texas ARP grant 14-748779.

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 1–2, 2007.

His current research interests include wireless sensor networks, smart environments, security, mobile and pervasive computing, resource and mobility management in wireless networks, mobile Internet, mobile grid computing, biological networking, applied graph theory and game theory. He has published over 400 papers in international conferences and journals, and over 30 invited book chapters. He holds five US patents in wireless networks and mobile Internet, and coauthored the book "Smart Environments: Technology, Protocols, and Applications" (John Wiley, 2005).

Dr. Das is a recipient of Best Paper Awards in IEEE PerCom'06, ACM MobiCom'99, ICOIN'02, ACM MSwiM'00 and ACM/IEEE PADS'97. He is also a recipient of UTA Academy of Distinguished Scholars Award (2006), University Award for Distinguished Record of Research (2005), College of Engineering Research Excellence Award (2003), and Outstanding Faculty Research Award in Computer Science (2001 and 2003). He is frequently invited as keynote speaker at international conferences and symposia.

Dr. Das serves as the Founding Editor-in-Chief of Pervasive and Mobile Computing (PMC) journal (Elsevier), and Associate Editor of IEEE Transactions on Mobile Computing, ACM/Springer Wireless Networks, IEEE Transactions on Parallel and Distributed Systems, and Journal of Peer-to-Peer Networking. He is the founder of IEEE WoWMoM and co-founder of IEEE PerCom conference. He has served as General or Technical Program Chair as well as TPC member of numerous IEEE and ACM conferences. He serves on IEEE TCCC and TCPP Executive Committees.

References

1. Liu, Y., Das, S.K.: Information Intensive Wireless Sensor Networks: Potential and Challenges. IEEE Communications 44(11), 142–147 (2006)
2. Luo, H., Liu, Y., Das, S.K.: Routing Correlated Data in Wireless Sensor Networks: A Survey, IEEE Network (to appear, November/December 2007)
3. Luo, H., Liu, Y., Das, S.K.: Routing Correlated Data with Fusion Cost in Wireless Sensor Networks. IEEE Transactions on Mobile Computing 5(11), 1620–1632 (2006)
4. De, P., Liu, Y., Das, S.K.: An Epidemic Theoretic Framework for Evaluating Broadcast Protocols in Wireless Sensor Networks. In: MASS. Proc. of IEEE Conference on Mobile Ad hoc and Sensor Systems, Pisa, Italy (October 2007)
5. Zhang, W., Das, S.K., Liu, Y.: A Trust Based Framework for Secure Data Aggregation in Wireless Sensor Networks. In: SECON. Proc. of Annual IEEE Conference on Sensor, Mesh, and Ad Hoc Communications and Networks, pp. 60–69 (September 2006)
6. Wu, X., Chen, G., Das, S.K.: The Energy Hole Problem in Wireless Sensor Networks with Nonuniform Node Distribution and Constant Data Reporting. IEEE Transactions on Parallel and Distributed Systems (to appear, 2008)
7. Ghosh, Das, S.K.: A Distributed Greedy Algorithm for Connected Sensor Cover in Dense Sensor Networks. In: DCOSS. Proc. of IEEE International Conference on Distributed Computing in Sensor Systems, pp. 340–353 (June 2005)
8. Choi, W., Das, S.K.: A Novel Framework for Energy-Conserving Data Gathering in Wireless Sensor Networks. In: Proc. of IEEE INFOCOM (March 2005)

QoS-Aware Cooperative and Opportunistic Scheduling Exploiting Multi-user Diversity for Rate Adaptive Ad Hoc Networks

Zhisheng Niu

Dept. of Electronic Engineering, Tsinghua University, Beijing 100084, China
niuzhs@tsinghua.edu.cn

Abstract. The recent researches in wireless networks prompt the opportunistic transmission that exploiting channel fluctuations to improve the overall system performance. In wireless ad hoc networks, nodes may have packets destined to multiple neighboring nodes. We consider an opportunistic scheduling that takes advantage of time-varying channel among different receivers to improve system performance. Maximizing overall throughput and satisfying QoS requirements for transmission flows are two important objectives that need to be considered. In literature, many opportunistic scheduling policies for ad hoc networks have been proposed, in which each transmitter schedules the transmission independently. However, due to co-channel interference, the decisions of neighboring transmitters are highly correlated. Moreover, to achieve the QoS requirements, nodes have to be cooperative to share the common wireless channel. In this paper, we formulate the opportunistic scheduling problem taking the interaction among the neighboring transmitters into account. We present an optimal scheduling policy which maximizes the overall network performance while satisfying QoS requirements of individual flows. We also proposed COS, a distributed Cooperative and Opportunistic Scheduling algorithm, which modifies IEEE 802.11 protocol to implement the optimal scheduling policy by exchanging average channel conditions and QoS factors among 2-hop neighboring nodes. Simulation results indicate that our implementation achieves higher network throughput and provides better QoS support than existing work.

Biography

Prof. Zhisheng Niu graduated from Northern Jiaotong University (currently Beijing Jiaotong University), Beijing, China, in 1985, and got his M.E. and D.E. degrees from Toyohashi University of Technology, Toyohashi, Japan, in 1989 and 1992, respectively. In 1994, he joined with Tsinghua University, Beijing, China, where he is now a full professor at the Department of Electronic Engineering. He is also an adjunction professor of Beijing Jiaotong University. From April 1992 to March 1994, he was with Fujitsu Laboratories Ltd., Kawasaki, Japan. From October 1995 to February 1996, he was a visiting research fellow of the Communications Research Laboratory of the Ministry of Posts and Telecommunications of Japan. From February 1997 to February 1998, he was a visiting senior researcher of Central

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 3–4, 2007.

Research Laboratory, Hitachi Ltd. He received the PAACS Friendship Award from the Institute of Electronics, Information, and Communication Engineers (IEICE) of Japan in 1991 and the Best Paper Award (1st prize) from the 6th Chinese Youth Conference on Communication Technology in 1999. He also received the Japanese Government Research Awards for Foreign Specialists from Science and Technology Agency (STA) of Japan and the Award for the Telecommunications Advancement Research Fellowship from Telecommunication Advancement Organization (TAO) of Japan in 1995. His current research interests include teletraffic theory, wireless/ mobile ATM/IP, radio resource management of wireless networks, and High Altitude Stratospheric Platforms (HAPS). He has published more than 100 journal and conference papers so far. Dr. Niu is a fellow of the IEICE, a senior member of the IEEE, Vice Director of Asia-Pacific Board and Beijing Chapter Chair of IEEE Communication Society, and member of Chinese Institute of Electronics Council.

Modeling the Effect of Forwarding in a Multi-hop Ad Hoc Networks with Weighted Fair Queueing

Ralph El Khoury and Rachid El-Azouzi

LIA/CERI, Université d'Avignon, Agroparc BP 1228, Avignon, France

Abstract. Consider a wireless ad hoc network with random access channel. We present a model that takes into account topology, routing, random access in MAC layer (governed by IEEE 802.11orslotted aloha) and forwarding probability. In this paper , we are focusing to study the effect of cooperation on the stability and throughput of ad-hoc network. Forwarding packets of other nodes is an example of activity that requires such a collaboration. Hence, it may not be in interest of a node to always forward the requesting packet. We propose a new approach (based on *cycle of transmissions*) to derive throughput of multi-hop routes and stability of forwarding queues. With this cycle approach, we correct the analytical expressions derived in [2] and discover that their results are valid only in particular cases such as symmetric networks. However, in this paper, we get extended results for general network case. Moreover, we confirm that (i) the forwarding queues in a system of weighted fair queues has a special property and (ii) the end to end throughput of a connection does not depend on the load of the intermediate forwarding queues between a source and a destination. We perform extensive simulations and verify that the analytical results exactly match the results obtained from simulations.

1 Introduction

A multi-hop wireless ad hoc network is a collection of nodes that communicate with each other without any established infrastructure or centralized control. Each of these nodes is a wireless transceiver that transmits and receive at a single frequency band which is common to all the nodes. These nodes can communicate with each other, however, they are limited by their transmitting and receiving capabilities. Therefore, they cannot directly reach all of the nodes in the network as most of the nodes are outside of direct range. In such a scenario, one of the possibilities for the information transmission between two nodes that are not in position to have a direct communication is to use other nodes in the network. To be precise, the source device transmits its information to one of the devices which is within transmission range of the source device. In order to overcome this, the network operates in a multi-hop fashion. Nodes route traffic for each other. Therefore, in a connected ad hoc network, a packet can travel from any source to its destination either directly, or through some set of intermediate packet forwarding nodes.

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 5–18, 2007.

Clearly, a judicious choice is required to decide on the set of devices to be used to assist in the communication between any two given pair of devices. This is the standard problem of routing in communication networks. The problem of optimal routing has been extensively studied in the context of wire-line networks where usually a shortest path routing algorithm is used: Each link in the network has a weight associated with it and the objective of the routing algorithm is to find a path that achieves the minimum weight between two given nodes. Clearly, the outcome of such an algorithm depends on the assignment of the [illegible] associated to each link in the network. In the wire-line context, there are many well-studied criteria to select these weights for links, such as delays. In the context of wireless ad-hoc networks, however, not sufficient attempts have been made to (i) identify the characteristics of the quantities that one would like to associate to a [illegible] as its weight, and in particular (ii) to understand the resulting network performance and resource utilization. Some simple heuristics have been frequently reported to improve performance of applications in mobile ad-hoc networks (see [10] and reference therein).

To study this problem, we consider in this paper the framework of random access mechanism for the wireless channel where the nodes having packets to transmit in their transmit buffers attempt transmissions by delaying the transmission by a random amount of time. This mechanism acts as a way to avoid collisions of transmissions of nearby nodes in the case where nodes can not sense the channel while transmitting (hence, are not aware of other ongoing transmissions). CSMA/CA mechanism in DCF is an example of this mechanism. When a station has frame to transmit, it sense the medium first. After the medium is sensed idle for a time interval (DIFS), it starts to transmit the frame. If a acknowledgement (ACK) frame is not received in SIFS, the station assumes the frame has experienced a collision and differs its transmission according to a Binary Exponential Backoff (BEB). We assume that time is slotted into fixed length time frames. In slotted aloha, the nodes decides with some fixed (possibly node dependent) probability in favor of a transmission attempt. If there is no other transmission by the other devices whose transmission can interfere with the node under consideration, the transmission is

At any instant in time, a device may have two kinds of packets to be transmitted:

1. Packets generated by the device itself. This can be sensed data if we are considering a sensor network.
2. Packets from other neighboring devices that need to be [illegible].

In this paper we consider two separate queues for these two types and do a weighted fair queueing (WFQ) for these two queues. This type of configuration allow us to include in the model the cooperation level which represents the fraction of the traffic forwarded by a node in ad-hoc network.

Several studies have focused on wireless network stability and finding the maximum achievable throughput. Among the most studied stability problems are scheduling [12, 13] as well as for the Aloha protocol [1, 11, 15]. Tassiulas and Ephremides [12] obtain a scheduling policy for the nodes that maximises the stability region. Their approach inherently avoids collisions which allows to maximize the throughput. Radunovic and Le Boudec [6] suggest that considering the total throughput as a performance objective may not be a good objective. Moreover, most of the related studied do not consider the problem of forwarding and each flow is treated similarly (except for Radunovic and Le Boudec [6], Huang and Bensaou [9] or Tassiulas and Sarkar [14]). Our setting is different than the mentioned ones in the following: the number of retransmission is finite, and therefore in our setting, the output and the input rates need not be the same. In recent past year, there has been a considerable effort on trying to increase the performance of wireless ad hoc networks since Gupta and Kumar [8] showed that the capacity of a fixed wireless network decreses as the number of nodes increases. Grossglauser and Tse [7] presented a two-phase packet forwarding technique for mobile ad-hoc networks, utilizing the multiuser diversity, in which a source node transmits a packet to the destination when this destination becomes the closet neighbors of the relay. This scheme was shown to increase the capacity of the MANET, such that it remains constant as the number of users in the MANET increases.

In [2], working with the above mentioned system model, we have already studied the impact of routing, channel access rates and weights of the weighted fair queueing on throughput, stability and fairness properties of the network. In this paper, we obtained important insights into various tradeoffs that can be achieved by varying certain network parameters. The main contribution of this paper is to provide approximation expression of stability. It states that whether or not the forwarding queues can be stabilized (by appropriate choice of WFQ weights) depends only on the routing and the channel access rates of the nodes. Further, in the stability region, the end-to-end throughput of a connection does not depend on the load on the intermediate nodes. In this paper, we extend the results in [2] to 802.11, and investigate the impact of forwarding probability on network stability.

The remainder of the paper is organized as follows. In section 2, we present the cross-layer network model with the new approach of [illegible]. In section 3, we write the balance rate equations from which appears the property of the forwarding queues and the throughput independency from the forwarding weight of the WFQ. The validation of analytical results is done with a discrete time simulator in section 4. Finally, we end with a conclusion in section 5.

2 Network Model

We model the ad hoc wireless network as a set of N nodes deployed arbitrarily in a given area. We number the nodes with integer numbers.

We assume the following:

- **A one simple channel:** Nodes use the same frequency for transmitting with an omni-directional antennas. A node j receives successfully a packet from a node i if and only if there is no interference at the node j due to another transmission on the same channel. A node cannot receive and transmit at the same time.

- **Two types of queues:** Two queues are associated with each node. The first one is the forwarded queue, noted by F_i (proper to the node i), which carry all the packets originated from a given source and destined to a given destination. The second is Q_i which carries the proper packets of the node i (in this case $i \equiv s$ where s designates a source node). We assume that each node has an infinite capacity of storage for the two queues. Packets are served with a first in first served fashion. When F_i has a packet to be sent, the node chooses to send it from F_i with a probability f_i. In other terms, it chooses to send from Q_i with probability $1 - f_i$. When one of these queues is empty then we choose to send a packet from the none empty queue with a probability 1. When node i decides to transmit from the queue Q_i, it sends a packet destined for node d, $d \neq i$, with probability $P_{i,d}$. The packets in each of the queues Q_i and F_i are served in first come first served fashion.
- **Saturated network:** Each node has always packets to be sent from queue Q_i, whereas F_i can be empty. Consequently, the network is considered saturated and depends on the channel access mechanism.

Network layer handles the two queues Q_i and F_i using the WFQ scheme, as described previously. Also, this layer maintains routing algorithms. So, each node acts as a router, it permits to relay packets originated from a source s to a destination d. It must carries a routing information which permits sending of packets to a destination via a neighbor. In this paper, we assume that nodes form a static network where routes between any source s and destination d are invariant in the saturated network case. Proactive routing protocols as OLSR (Optimized Link State Routing) construct and maintain a routing table that carry routes to all nodes on the network. These kind of protocols correspond well with our model. Note that the set of nodes between a node s and d is designated by $R_{s,d}$. Let $R_{i,s,d}$ be the set of nodes $R_{s,i} \bigcup i$.

We assume that all nodes use the same transmit power T, hence the signal T_r received at the receiver is a decreasing function of the distance D and given by $T_r = \frac{gT}{D^\alpha}$. The receiver can correctly decode the signal from a node s if the signal to interference ration (SIR) exceeds a certain threshold β, i.e.,

$$SIR = \frac{T_r}{\sum_{s' \neq s} T/d^\alpha} \geq \beta$$

where D is the distance between the receiver and the source. We define the neighbors set $\mathcal{N}(i)$ of a receiver i as the set of nodes whose transmission, if overlapping with the transmission of sender s, will cause collision at the receiver, i.e

$$\mathcal{N}(i) = \{s' / \frac{gT/D^\alpha}{gT/\bar{D}^\alpha}\}$$

where $\bar{D}$ is the distance between the receiver i and the source s'. Without loss of generality, we assume that the carrier sense set of sender is coincide with its

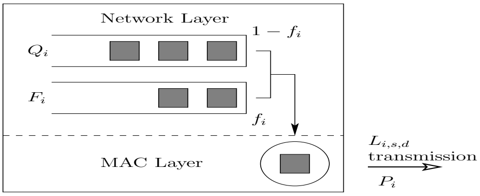


Fig. 1. Network layer and MAC layer of node i

neighbors set. That is means the sender will sense the channel to be busy if any of neighbors transmit.

In the MAC layer, we assume a channel access mechanism only based on a probability to access the network i.e. when a node i has a packet to transmit from the queue Q_i or F_i, it accesses the channel with a probability P_i. It can be similar to CSMA/CA or any other mechanism to access the channel. For example, in IEEE 802.11 DCF, the transmission probability or attempt probability is given by [5]

$$P = \frac{2(1 - 2P_{coll})}{(1 - 2P_{coll})(CW_{min} + 1) + P_{coll}CW_{min}(1 - (2P_{coll})^m)}, \tag{1}$$

where P_{coll} is the conditional collision probability given that a transmission attempt is made, and $K = log_2(\frac{CW_{max}}{CW_{min}})$ is the maximum of backoff stage.

In this paper, we allow that the nodes have different backoff. The above definitions capture the possibility of having different CW_{min} and CW_{max} values, different exponential backoff multiplier values and different number of permitted attempts. The computation of attempt rate is given in [4].

We have considered previously infinite buffer size, therefore, there is no packet loss due to overflow at the queues. The only source of packet loss is due to collisions. For a reliable communication, we allow a limit number of successive transmissions of a single loosed packet, after that it will be dropped definitively. We denote $K_{i,s,d}$ the maximum number of successive collisions allowed of a single packet sent from the node i on the path from s to d. $K_{i,s,d}$ is known as the limit number of retry. Let also $L_{i,s,d}$ be the expected number of attempts till successful or a definitively drop from node i on the path from s to d.

2.1 Cross-Layer Representation of the Model

The model of figure 1 represents our model in this paper. The two layers are clearly separated. Attempting the channel begins by choosing the queue from which a packet must be selected. And then, this packet is moved from the corresponding queue from the network layer to the MAC layer where it will be transmitted and retransmitted, if needed, until its success or drop. In this manner, when a packet is in the MAC layer, it is itself attempted successively until it is removed from the node.

We present the model for each node i, as an axe of time graduated by time slots and cycles . A cycle is defined as the number of slots needed to transmit a single packet until its success or drop. We distinguish two types of cycles: The [illegible] in relief to the packets of F_i and the [illegible] in relief to the packets coming from Q_i. Also, each cycle is affected to a connection. The beginning of each cycle represents the choice of the queue from which we choose a packet and the choice of the connection where to send it. Whereas, the slots that constitute the cycle represents the attempts of the packet itself to the channel, including its retransmissions. Hence, the distinction of the network and MAC layer is now clear.

We need to define formally the model, so we will be able to derive some formulas in the next sections. For that, consider the following counters:

- $C_{t,i}$ is the number of cycle of the node i till the t^{th} slot.
- $C^F_{t,i}$ (resp. $C^Q_{t,i}$) is the number of all [illegible] (resp. [illegible]) of the node i till the t^{th} slot.
- $C^F_{t,i,s,d}$ (resp. $C^Q_{t,i,s,d}$) is the number of [illegible] (resp. [illegible]) corresponding to the path $R_{s,d}$ of the node i till the t^{th} slot.
- $T_{t,i,s,d}$ is the number of times we found at the first slot of a cycle and at the first position in the queue F_i a packet for the path $R_{s,d}$ of the node i till the t^{th} slot.
- $I_{t,i,s,d}$ is the number of cycles corresponding to the path $R_{s,d}$ of the node i, and where a cycle is ended by a success of the transmitted packed, till the t^{th} slot.
- $A_{t,i,s,d}$ is defined with: $A_{t,j_{i,s,d},s,d} = I_{t,i,s,d}$ which is the arrival of successful packets from node i to its next hop $j_{i,s,d}$ on the path $R_{s,d}$.

Figure 2 shows a simple example with some numerical values of the previous counters for a single node i.

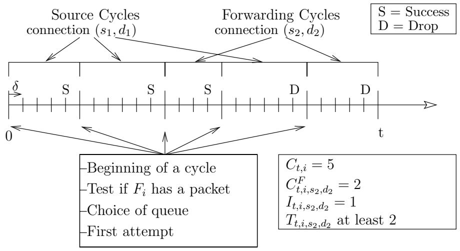


Fig. 2. Illustrative example of node i with cycles approach

3 Stability Properties of the Forwarding Queues

Our main objective in this section is to derive the rate balance equations from which some properties of the forwarding queues can be deduced. For that we need to write the departure rate from each node i and the end to end throughput between a couple of node.

From a practical point of view, each node owns three main parameters P_i, $K_{i,s,d}$ and f_i, that can be managed and set in such a way that each node can maintain stability, or the end to end throughput on a path can be optimized. In this paper, by fixing the routing paths (from $R_{s,d}$) and route choice (from $P_{s,d}$), we will obtain forwarding queue stability function of these three main parameters.

For a given routing, let π_i denote the probability that the queue F_i has at least one packet to be forwarded in the beginning of each cycle. $\pi_{i,s,d}$ is the probability that the queue F_i has a packet at the first position ready to be forwarded to the path $R_{s,d}$ in the beginning of each cycle. Let n_i be the number of neighboring nodes of node i.

Giving a saturated network case where each node has a packet on its queue Q_i and attempts transmitting all the time to the channel, the forwarding queue F_i of each node will have a π_i load when it tries to forward packets to its neighbors.

3.1 The Rate Balance Equations

The forwarding queue F_i is stable if the departure rate of packets from F_i is equal to the arrival rate into it. This is a simple definition of stability that can be written with a [illegible]. In this paper, we are going to derive this equation for each node i using the cycle approach of the model 2 (figure 1). In fact, it is judicious to write the rate balance equation of node i for each connection and then do the summation for all others.

For any given nodes i, s and d, let $j_{i,s,d}$ be the entry in the set $R_{s,d}$ just after i. It is possible that there is no such entry, i.e., node i is the last entry in the set $R_{s,d}$. In that case $j_{i,s,d} = d$. Let $P_{i,s,d} = \Pi_{j \in j_{i,s,d} \cup \mathcal{N}(j_{i,s,d}) \setminus i}(1 - P_j)$ be the probability that a transmission from node i on route from node s to node d is successful. Also, let

$$L_{i,s,d} = \sum_{l=1}^{K_{i,s,d}} l(1 - P_{i,s,d})^{l-1} P_{i,s,d} + K_{i,s,d}(1 - P_{i,s,d})^{K_{i,s,d}} = \frac{1 - (1 - P_{i,s,d})^{K_{i,s,d}}}{P_{i,s,d}} \tag{2}$$

be the expected number of attempts till success or consecutive $K_{i,s,d}$ failures of a packet from node i on route $R_{s,d}$.

The Departure Rate. The probability that a packet is removed from a node i by a successful transmission or a drop (i.e. a successive $K_{i,s,d}$ failure) is the departure rate from F_i. We denote it by d_i. The departure rate concerning only the packets sent on the path $R_{s,d}$ is denoted $d_{i,s,d}$. Formally, for any node i, s and d such that $P_{s,d} > 0$ and $i \in R_{s,d}$, the long term departure rate of packets from node i on the route from s to d is

$$d_{i,s,d} = \lim_{t \to \infty} \frac{C^F_{t,i,s,d}}{t} = \lim_{t \to \infty} \frac{T_{t,i,s,d}}{C_{t,i}} . \frac{C^F_{t,i,s,d}}{T_{t,i,s,d}} . \frac{C_{t,i}}{t} \tag{3}$$

- $\frac{T_{t,i,s,d}}{C_{t,i}}$ is exactly the probability that F_i carried a packet to the path $R_{s,d}$ in the beginning of each cycle. Therefore, $\frac{T_{t,i,s,d}}{C_{t,i}} = \pi_{i,s,d}$.
- $\frac{C^F_{t,i,s,d}}{T_{t,i,s,d}}$ is exactly the probability that we have chosen a packet from F_i to be sent when F_i carried a packet to the path $R_{s,d}$ in the first position and in the beginning of a forwarding cycle. Therefore, $\frac{C^F_{t,i,s,d}}{T_{t,i,s,d}} = f_i$.
- $\frac{t}{C_{t,i}}$ is the average length in slots of a cycle of the node i. A cycle length on the path $R_{s,d}$ is formed by the attempt slots that does not lead to a channel access and the transmission and retransmissions of the same packet until a success or a drop. Thus a cycle length for a one path $R_{s,d}$ of a node i is $\frac{L_{i,s,d}}{P_i}$. When a node transmits to several paths, we need to know the average cycle length. This given by $\frac{\overline{L_i}}{P_i}$ where $\overline{L_i}$ is the average of $L_{i,s,d}$s of these paths. $\overline{L_i}$ is given by:

$$\overline{L_i} = \sum_{s,d:i\in R_{sd}} \pi_{i,s,d} f_i L_{i,s,d} + \sum_d (1 - \pi_i f_i) P_{i,d} L_{i,i,d} \tag{4}$$

Therefore, $\frac{C_{t,i}}{t} = \frac{P_i}{\overline{L_i}}$. Consequently, $d_{i,s,d} = \pi_{i,s,d} f_i \frac{P_i}{\overline{L_i}}$.

It is clear that the the departure rate $d_{i,s,d}$ on a path $R_{s,d}$ of a node i does not depend on the parameters of only one path but it is also related to the expected number of transmissions to all other paths used by node i. This dependency appears in $\overline{L_i}$ which is not introduced in paper [2]. Moreover, it is easy to derive the total departure rate d_i on all paths:

$$d_i = \sum_{s,d:i\in R_{s,d}} d_{i,s,d} = \pi_i f_i \frac{P_i}{\overline{L_i}} \tag{5}$$

The Arrival Rate. The probability that a packet arrives to the queue F_i of the node i is the arrival rate that we denoted it by a_i. When this rate concerns only packets sent on the path $R_{s,d}$, we denoted it by $a_{i,s,d}$. Formally, for any node i, s and d such that $P_{s,d} > 0$ and $i \in R_{s,d}$, the long term arrival rate of packets into F_i for $R_{s,d}$ is

$$a_{i,s,d} = \lim_{t\to\infty} \frac{A_{t,i,s,d}}{t} = \lim_{t\to\infty} \frac{C^Q_{t,s}}{C_{t,s}} . \frac{C^Q_{t,s,s,d}}{C^Q_{t,s}} . \frac{C_{t,s}}{t} . \frac{I_{t,s,s,d}}{C^Q_{t,s,s,d}} . \frac{A_{t,i,s,d}}{I_{t,s,s,d}} \tag{6}$$

- $\frac{C^Q_{t,s}}{C_{t,s}} = 1 - \frac{C^F_{t,s}}{C_{t,s}} = 1 - \pi_s f_s$, this is exactly the probability to get a source cycle i.e. to send a packet from the queue Q_s.
- $\frac{C^Q_{t,s,s,d}}{C^Q_{t,s}}$ is the probability to choose the path $R_{s,d}$ to send a packet from Q_s. Therefore, $\frac{C^Q_{t,s,s,d}}{C^Q_{t,s}} = P_{s,d}$.
- $\frac{C_{t,s}}{t} = \frac{P_s}{\overline{L_s}}$.

- $\frac{I_{t,s,s,d}}{C^Q_{t,s,s,d}}$ is the probability that a source cycle on the path $R_{s,d}$ ends with a success i.e. the packet sent from Q_s is received on the queue $F_{j_{s,s,d}}$. Therefore, $\frac{I_{t,s,s,d}}{C^Q_{t,s,s,d}} = (1-(1-P_{s,s,d})^{K_{s,s,d}})$.
- $\frac{A_{t,i,s,d}}{I_{t,s,s,d}}$ is the probability that a packet received on the node $j_{s,s,d}$ is also received on the queue F_i of the node i. For that, this packet needs to be received by all the nodes in the set $R_{i,s,d}$. Therefore, $\frac{A_{t,i,s,d}}{I_{t,s,s,d}} = \prod_{k \in R_{i,s,d} \setminus i}(1-(1-P_{k,s,d})^{K_{k,s,d}})$.

Consequently,

$$a_{i,s,d} = (1-\pi_s f_s).P_{s,d}.\frac{P_s}{L_s}.\Big[(1-(1-P_{s,s,d})^{K_{s,s,d}}).\prod_{k \in R_{i,s,d} \setminus i}(1-(1-P_{k,s,d})^{K_{k,s,d}})\Big]$$

Remark that when the node i is the destination of a path $R_{s,d}$, then $a_{d,s,d}$ represents the end to end average throughput of a connection from s to d. Also, note that the global arrival rate is: $a_i = \sum_{s,d: i \in R_{sd}} a_{i,s,d}$

The rate balance. Finally, in the steady state if all the queues in the network are stable, then for each i, s and d such that $i \in R_{s,d}$ we get $d_{i,s,d} = a_{i,s,d}$, which is the rate balance equation on the path $R_{s,d}$.

Let $y_i = 1 - \pi_i f_i$ and $z_{i,s,d} = \pi_{i,s,d} f_i$. Thus $y_i = 1 - \sum_{s,d: i \in R_{sd}} z_{i,s,d}$. Then rate balance equation becomes,

$$\sum_{d: i \in R_{s,d}} z_{i,s,d} = \frac{y_s(\sum_{s',d'} z_{i,s',d'} L_{i,s',d'} + \sum_{d"} y_i P_{i,d"} L_{i,i,d"})w_{s,i}}{(\sum_{s',d'} z_{s,s',d'} L_{s,s',d'} + \sum_{d"} y_s P_{s,d"} L_{s,s,d"})} \tag{7}$$

where

$$w_{s,i} = \sum_{d: i \in R_{s,d}} \frac{P_{s,d} P_s}{P_i} \prod_{k \in R_{i,s,d} \cup s \setminus i} (1-(1-P_{k,s,d})^{K_{k,s,d}})$$

3.2 Interpretation and Applications

The relation of equation (7) has many interest interpretation and applications. Some of these are :

- [illegible] f_i At the heart of all the following points is the observation that the quantities $z_{i,s,d}$ and y_i are [illegible] of the choice of f_j, $1 \leq j \leq N$. It only depends on the routing and the value of P_j.
- [illegible] Since the values of y_i are independent of the values of f_j, $j = 1, \ldots, N$, and since we need $\pi_i < 1$ for the forwarding queue of node i to be stable, we see that for any value of $f_i \in (1-y_i, 1)$, the forwarding queue of node i will be stable. Thus we obtain a lower bound on the weights given to the forwarding queues at each node in order to guarantee stability of these queues. To ensure that these lower bounds are all feasible, i.e., are less than 1,

we need that $0 < y_i \leq 1$; $y_i = 0$ corresponds to the case where F_i is unstable. Hence, if the routing, $P_{s,d}$ and $P_j s$ are such that all the y_i are in the interval $(0, 1]$, then all the forwarding queues in the network [illegible] f_i. Now, since y_i is determined only by routing and the probabilities $P_j s$ and $P_{s,d}$, we can then [illegible] f_i (thereby also fixing π_i, hence the forwarding delay) to satisfy some further optimization criteria so that this extra degree of freedom can be exploited effectively.

- [illegible] We see that the long term rate at which node s can serve its own data meant for destination d is $P_{s,d}P_s(1 - \pi_s f_s) = P_{s,d}P_s y_s$ which is [illegible] f_s. The throughput, i.e., the rate at which data from node s reaches their destination d which is given by $a_{d,s,d}$, turns out to be independent of the choice of $f_j, 1 \leq j \leq N$. Similarly, the long term rate at which the packets from the forwarding queue at any node i are attempted transmission is $P_i \pi_i f_i = P_i(1 - y_i)$, which is also independent of the choice of $f_j, 1 \leq j \leq N$.
- [illegible] f_i Assume that we restrict ourselves to the case where $f_i = P_f$ for all the nodes. Then, for stability of all the nodes we need that

  $$P_f > 1 - \min_i y_i.$$

  Since the length of the interval that f_i is allowed to take is equal to y_i, we will also refer to y_i as stability region.
- [illegible] For a given set of P_js, $P_{s,d}$ and routing, the throughput obtained by any route $R_{l,m}$ is fixed, independent of the forwarding probabilities f_i. Hence there is no [illegible] tradeoff that can be obtained by changing the forwarding probabilities. A real tradeoff is caused by the maximum number of attempts: the throughput is ameliorated when reattempting many times on a path, while the service rate on a forwarding queue is slowed down causing low stability region and delay will be increased.
- [illegible] Note that the value $\pi_i f_i E_r$ represents the energy consumption used by node i to forward the packets of other connections where E_r is the energy spent for transmission of one packet. This quantity turns out to be independent of the choice of f_i. Hence, the node can use f_i to improve the expected delay without affecting the energy consumption.

4 Numerical Results and Simulations

In this section, we present some numerical results and validate the expressions found in the previous sections with a discrete time network simulator. We have implemented this simulator according to the model of section 2. Hence, it appears to be a valuable tool of measurement. We deploy an asymmetric static wireless network with 11 nodes as shown in figure 3.

Five connections are established a, b, c, d and e as indicated in the same figure (a dashed or complete line between two nodes in this figure means that there is

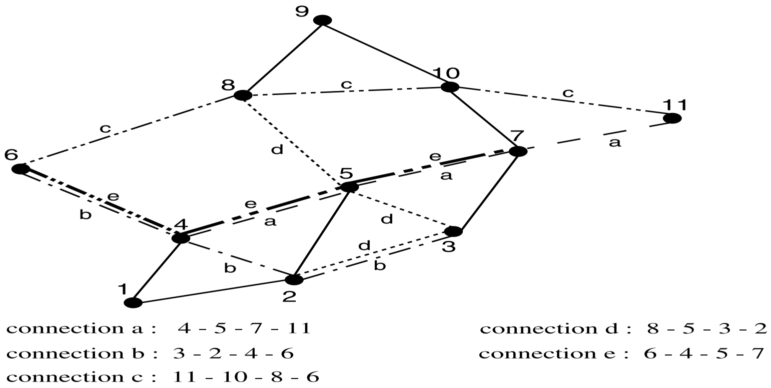
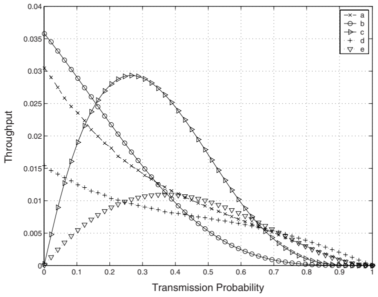

9
8
10
11
6
5
7
4
3
1
2
a
b
c
d
e
connection a : 4 - 5 - 7 - 11
connection b : 3 - 2 - 4 - 6
connection c : 11 - 10 - 8 - 6
connection d : 8 - 5 - 3 - 2
connection e : 6 - 4 - 5 - 7

Fig. 3. Wireless network



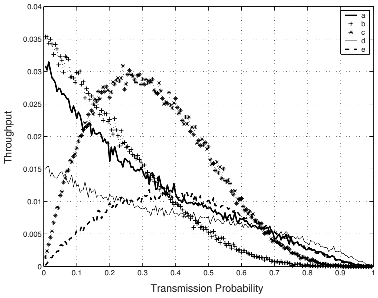


Fig. 4. Throughput from analytical model for $K = 4$ and $f = 0.8$

Fig. 5. Throughput from simulation for $K = 4$ and $f = 0.8$

a neighboring relation). These connections choose the shortest path in terms of hops to route their packets. We choose the parameters $K_{i,s,d} \equiv K$, $f_i \equiv f$ and P_i in a manner of enabling stability, for all i, s and d. We fix $f = 0.8$ except contraindication. Let $P_2 = 0.3, P_3 = 0.3, P_4 = 0.4, P_5 = 0.5, P_7 = 0.3, P_8 = 0.3, P_{10} = 0.4$ be the fixed transmission probabilities for nodes 2, 3, 4, 5, 7, 8 and 10 while $P_i \equiv P$ for all other i. Many nodes need to have a fix transmission probabilities so to get a stable queues for all nodes.

First, we validate the analysis results via simulation. A simulated Ad-Hoc network scenario is configured to study our analytical model. In figures 4 and 6 (resp. 5 and 7), we plot the throughput computed by analytical model (resp. simulation) on various routes and the quantities π_i versus the channel access probability for $K = 4$. We observe that the analytical results match closely the simulator result.

In figures 8 and 9, we plot the throughput on various routes and the probability π_i versus the transmission probability. The existence of an optimal channel access rate (or, the transmission probability) is evident from the figures.

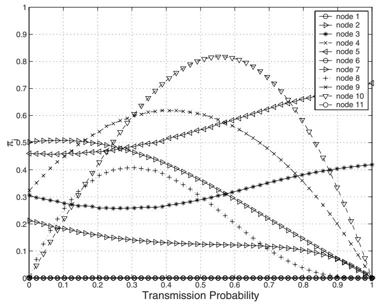


Fig. 6. π_i from analytical model for $K = 4$ and $f = 0.8$

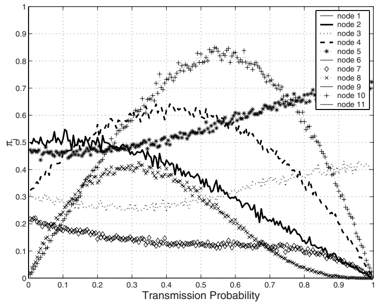


Fig. 7. π_i from simulation for $K = 4$ and $f = 0.8$

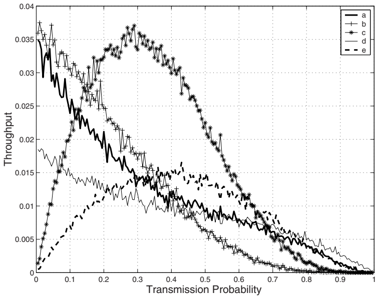


Fig. 8. Throughput from simulation for $K = 5$ and $f = 0.8$

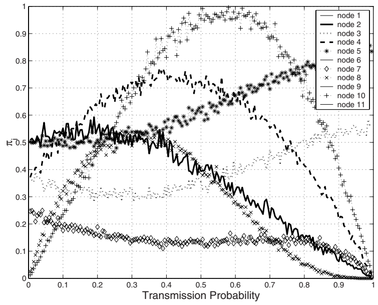


Fig. 9. π_i from simulation for $K = 5$ and $f = 0.8$

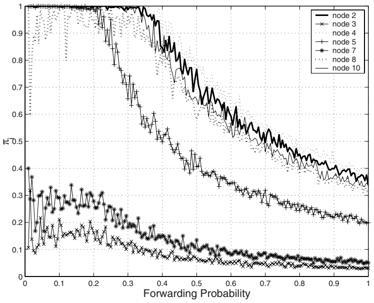


Fig. 10. π_i from simulation versus the forwarding probability for $K = 4$

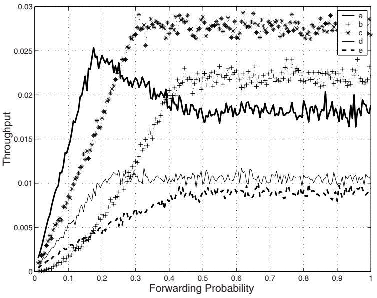


Fig. 11. Throughput from simulation for $K = 4$ as function of the forwarding probability

Moreover, as expected, the optimal transmission probability increases with K. The figures 4-9 show that increasing the parameter K significantly improves the throughput but the region of stability decreases. It is therefore clear, there is a throughput-stability tradeoff which can be obtained by changing the maximum number of transmission (K).

From figure 11 to 11, we vary the load of the forwarding queues by changing the forwarding probability of each node. The parameter of the network are as follow: $K = 4$, $P_2 = 0.5, P_3 = 0.3, P_4 = 0.45, P_5 = 0.5, P_7 = 0.3, P_8 = 0.3, P_{10} = 0.4$ and the rest of the nodes have $P = 0.2$. We observe that when f is small the system is not stable, more precisely the nodes 2, 4, 5, 8 and 10 are suffering from a congestion as shown in figure 10. They need to deliver more packets from the forwarding queue in a faster manner. In this unstable case, the throughput of all connections is sensitive with the f variation. As we see in figure 11, it increases with f until the system becomes stable around $f = 0.4$.

5 Conclusion

In this paper, we have presented a cross-layer model that takes into accounts many parameters concerning network and MAC layer. By separating clearly the role of each layer with the [illegible] approach , we correct the analytical expressions of [2] and discover that their results can be valid only in particular cases. Moreover, we confirm that forwarding queues does not depend on the forwarding probability which is the weight of the WFQ. As a consequence, the end to end throughput between a couple of nodes does not depend on the load of the intermediate forwarding queues. We have performed numerical results and simulations to validate our findings. A perspective of this work is to consider the unsaturated case and study analytically the impact of the forwarding probability on the performances of the network.

References

1. Anantharam, V.: The stability region of the finite-user slotted Alloha protocol. III Trans. Inform. Theory 37(3), 535–540 (1991)
2. Kherani, A., El Azouziet, R., Altman, E.: Stability-Throughput Tradeoff and Routing in Multi-Hop Wireless Ad-Hoc Networks. In: The proceeding of Networking Conference, Coimbra, Portugal (Best paper award) (MAY 15, 19 2006), Available online: http://www.lia.univ-avignon.fr/fich_art/678-ELAZOUZI.pdf
3. Bianchi, G.: Performance analysis of the IEEE 802.11 distribute coordination function. IEEE Journal on Selected Areas in Communications (March 2000)
4. El-Azouzi, R.: Perfomance analysis of 802.11e in the multi-hop wireless network (submitted)
5. Yang, Y., Hou, J., Kung, L.: Modeling of Physical Carrier Sense in Multi-hop Wireless Networks and Its Use in Joint Power Control and Carrier Sense Adjustment, Infcom, Alaska (2007)

6. Radunovic, B., Le Boudec, J.Y.: Joint Scheduling, Power Control and Routing in Symmetric, One-dimensional, Multi-hop Wireless Networks. In: WiOpt 2003: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, Sophia-Antipolis, France (March 2003)
7. Grossglauser, M., Tse, D.: Mobility Increases the Capacity of Adhoc Wireless Networks. IEEE/ACM Transactions on Networking 10(4), 477–486 (2002)
8. Gupta, P., Kumar, P.R.: The capacity of wireless networks. III Trans. Inform. Theory 46(2), 388–404 (2000)
9. Huang, X., Bensaou, B.: On Max-min fairness and scheduling in wireless Ad-Hoc networks: Analytical framework and implementation. In: Proceeding MobiHoc 2001, Long Beach, California (October 2001)
10. Bansal, S., Shorey, R., Kherani, A.: Performance of TCP and UDP Protocols in Multi-Hop Multi-Rate Wireless Networks. In: IEEE WCNC, Atlanta, USA (March 2004)
11. Szpankowski, W.: Stability condition for some multiqueue distributed systems: buffered random access systems. Adv. Appl. Probab. 26, 498–515 (1994)
12. Tassiulas, L., Ephremides, A.: Stability properties of constrained queuing systems and scheduling for maximum throughput in multihop radio network. IEEE Trans. Automat. Contr. 37(12), 1936–1949 (1992)
13. Tassiulas, L.: Linear complexity algorithm for maximum throughput in radio networks and input queued switches. IEEE Infocom 98, 533–539 (1998)
14. Tassiulas, L., Sarkar, S.: Max-Min fair scheduling in wireless networks. In: Proceeding of Infocom 2002 (2002)
15. Tsybakov, B.S., Bakirov, V.L.: Packet transmission in radio networks. Probl. Infom. Transmission 21(1), 60–76 (1985)

Localized Mobility Control Routing in Robotic Sensor Wireless Networks

Hai Liu[1], Amiya Nayak[1], and Ivan Stojmenović[2,1]

[1] School of Information Technology and Engineering, University of Ottawa, ON, Canada
[2] Electronic, Electrical & Computer Engineering, The University of Birmingham, United Kingdom
{hailiu,anayak,ivan}@site.uottawa.ca

Abstract. The paper addresses mobility control routing in robotic sensor wireless networks, where either mobile sensors or mobile actuators assist in wireless data intensive transmissions from sensors. Given a communication request between a source destination pair, the problem is to find a route and move each node on the route to its desired location, such that total transmission power is minimized. We study the optimal number of hops and optimal distance of adjacent nodes on the route. We propose OHCR algorithm that is based on the optimal number of hops on the route. We further propose MPoPR algorithm that minimizes transmission power over progress. After finding an initial path, strategies of move in rounds and move directly are both considered to move nodes to desirable locations. Extensive simulations are conducted to evaluate our proposed routing algorithms.

1 Introduction

Ad hoc and sensor wireless networks hold great promise for providing easy to use mobile services for many applications. As technology rapidly progresses, diverse sensing and mobility capabilities will become more readily available to devices. For example, many modern mobile robots are already equipped with various sensing capabilities. As another example, there are presently research activities on low-power robotic in a variety of ways, including flying and skimming across the surface of water [12]. Recent progress in two seemingly disparate research areas namely, distributed robotics and low power embedded systems has led to the creation of mobile (or robotic) sensor networks. Autonomous node mobility brings with it its own challenges, but also alleviates some of the traditional problems associated with static sensor networks [4]. There are some requirements on long-term communication for these mobile sensor networks. For example, sensors may perform transmissions of continuous image or video to sink, or some critical measurements like temperatures are required from sensors to sink.

A lot of works studied effects of mobility of nodes and often regarded it as a passive fact that causes link broken of routes, disconnection of network topologies, etc.. However, mobility of nodes can potentially be used to improve network performance. For example, mobility of nodes can be used to deploy sensors at optimal

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 19–31, 2007.

locations for monitoring [11], increase throughput while keeping the small delay [8, 2], apply to long-term data gathering, and be used in aggregation of large data events. All above works assumed that mobility of nodes follows some given model of movement, i.e., mobility is not actually controllable. However, we can see the potential ability of controlled mobility to improve network performance.

In this paper, we study mobility of nodes as one of controllable parameters of networks, such as transmission radius of nodes. We focus on the networks of mobile nodes that autonomously perform long-term communication tasks. In general, long-term communication tasks which exhibit persistent or habitual communication patterns are prime candidates for the application of mobility to improve network performance. In such situation, the traffic will be regular enough and large enough in volume to warrant nodes expending energy moving in order to forward traffic in a more energy efficient manner. For any given communication request between a source destination pair, our problem is to find a route between the source destination pair, and move each node on the route to its desired location. Our objective is to minimize the total transmission power of nodes for this long-term communication, while keeping low movement distances.

Current solution for the problem is to adopt some existing routing protocols to find an initial route, and iteratively move each node to the midpoint of its upstream node and downstream node on the route [7]. However, existing routing protocols may not be efficient in the situation that nodes are able to move. Moving strategy in [7] may cause unnecessary zig-zig movements of nodes. In this work, we first study the optimal number of hops and optimal distance of adjacent nodes on the route from source to destination. Two new routing algorithms are designed to facilitate mobility control of nodes. Not only move in rounds strategy, but also move directly strategy is studied in our proposed routing algorithms.

The rest of this paper is organized as follows. Section 2 is related work. We propose Optimal Hop Count Routing algorithm and Minimum Power over Progress Routing algorithm in section 3. Section 4 is simulation that consists of 5 parts. We conclude our work in section 5.

2 Related Work

Mobility has been extensively investigated in ad hoc and sensor wireless networks [8, 2, 11]. A self-deployment algorithm for mobile sensor networks was studied in [11]. It assumed nodes have locomotion capability which enables a node to move to its optimal deployment location. An incremental deployment algorithm was proposed to deploy nodes such that network coverage is maximized. Relationship between mobility and network capacity was studied in [8]. It supposed there are n nodes communicating in random source-destination pairs in the network. It has been showed that the per-user throughput increases dramatically when nodes are mobile rather than fixed. Not only throughput, but also delay guarantees were studied with mobility of nodes in [2]. It was showed that mobility is able to increase capacity of networks while keeping the small delay. In mentioned works, mobility is assumed to follow some given model of movement, such as the random way-point model. The mobility of nodes is not actually controllable.

Mobility was considered as a controllable parameter in [7], [5] and [15]. The problem of load balanced data collection in wireless sensor networks was studied in [15]. The load of sensor nodes can be more balanced if a base station changes its location from time to time. The basic idea of this work is to combine existing multi-hop routing protocols with mobility of the base station to achieve improvements in terms of network lifetime. Simulation results showed that the joint mobility and routing strategy achieves 500% improvement of the network lifetime. However, only the base station is assumed to be mobile in [15]. Mobility of robotic relay stations was studied to maintain communication between an explorer and a base station [5]. It assumed the base station is static while the explorer goes an arbitrary way. The basic idea is to organize relay stations as a straight line and move each relay station to the middle point of its neighbors on the line. Our problem is most related to the problem discussed in [7]. It assumed that nodes in the network are able to move to desirable locations in order to improve power efficiency for one unicast flow, multiple unicast flows, and many-to-one flows. Optimal configuration of relay nodes was studied. It was showed that the optimal locations of relay nodes must lie entirely on the line between the source and destination, and the relay nodes must be evenly spaced along the line. The routing scheme was proposed as follows: 1) apply some greedy routing protocol to establish an initial route; 2) iteratively, each node (except for source and destination) moves to the midpoint of its upstream node and downstream node on the route. Two representative routing algorithms that can be adopted are Greedy routing [6] and NP [18] (nearest with progress) routing. In Greedy algorithm, current node on the route selects the neighbor that is closest to the destination. In NP algorithm, current node selects the neighbor that is closet to itself. Only neighbors closer to the destination than current node are considered in both Greedy and NP algorithms. Otherwise, route failure will be reported. Simulation results showed that controlled mobility of nodes can significantly improve network performance in many scenarios. However, drawback of this routing scheme is obvious. First, the initial route computed by existing routing algorithms may not keep energy efficiency after movement of nodes. It will degrade the advantage of mobility control for reducing transmission power. Second, iterative movement of nodes in rounds not only requires synchronization in all nodes, but also causes unnecessary zig-zig movement of nodes. It results in large delay which may cause failure of communications in the end.

A survey in [14] shows there are three basic metrics that are commonly adopted in current energy-efficient routing algorithms. They are minimum energy metric, min-max metric and minimum cost metric. I. Stojmenovic proposed a general framework for designing network layer protocols for sensor networks including localized routing [17]. The framework is based on optimizing the ratio of the cost of making certain decisions (e.g., selecting a forwarding neighbor for routing) to the progress made in doing so (e.g., reduction in distance to destination). It was shown that this general guideline can be applied to the design of hop count, power awareness, maximal lifetime, beaconless and physical-layer-based routing. However, current routing algorithms may not keep energy efficiency after movement of nodes in our problem.

3 Mobility Control Routing

The network topology is represented by an undirected graph. Source reports results of some sensing task to destination. It is a long-term communication and traffic amount is large enough to warrant relay nodes expending energy moving for more energy efficient traffic. We take the same assumption in [7] that the source and the destination do not move while relay nodes are able to move to their desirable locations. We assume that all nodes have the common communication radius r, and there is an edge between two nodes iff their Euclidean distance is less than r. The widely used energy cost model, $d^{\alpha}+c$, is adopted, where d is communication distance, α is signal decline factor and c is constant [16, 9]. Each node in the network is assumed to know locations of its neighbors and its own. The location information of each node can be obtained via GPS [10] or some distributed localization methods [1] when GPS service is not available. Location information of neighbors can be obtained by exchanging HELLO messages periodically.

Given a communication request between a source destination pair, our solution takes two steps to set up a route from the source to the destination with mobility control. First, we compute optimal number of hops and optimal distance of adjacent nodes on the route. Second, we propose a routing algorithm that is based on the optimal number of hop counts, and a greedy routing algorithm that minimizes transmission power over progress in selecting a forwarding neighbor. Two steps are discussed one by one.

3.1 Optimal Number of Hops and Distance

Suppose a source, say s, needs to find a route to a destination, say t. According to [7], the optimal locations of relay nodes must lie entirely on the line between s and t, and the relay nodes must be evenly spaced along the line. Actually, the optimal number of hops and optimal distance of adjacent nodes on the line can be computed.

Following theorem shows the optimal number of hops and optimal distance of adjacent nodes on the route from s to t.

Theorem 1. Given locations of s and t, to minimize total transmission power of route from s to t, the optimal number of hops on the route is integer k, such that $|k-d(s,t)\times((\alpha-1)/c)^{1/\alpha}|$ is minimized, and the optimal distance of adjacent nodes is $d(s,t)/k$, provided energy cost model is $d^{\alpha}+c$.

Proof. Without loss of generality, suppose there are k hops on the line from s to t after movement. Note that the relay nodes must be evenly spaced along the line. The distance of adjacent nodes is $d(s,t)/k$, where $d(s,t)$ is the Euclidean distance between s and t. So the total transmission power of nodes on the line is

$$k\times((d(s,t)/k)^{\alpha}+c). \qquad (1)$$

Let $x=d(s,t)/k$, denote the distance of adjacent nodes on the line. (1) is represented as

$$d(s,t)\times(x^{\alpha}+c)/x. \qquad (2)$$

We calculate derivation on x in (2) to minimize the total transmission power of nodes on the line. We have $d(s,t)\times(\alpha-1)x^{\alpha-2}-d(s,t)\times c\times x^{-2}=0$. Thus, we get $x=(c/(\alpha-1))^{1/\alpha}$. The optimal number of hops, k, can be computed by rounding $d(s,t)\times((\alpha-1)/c)^{1/\alpha}$ to the nearest integer. So the optimal space between adjacent nodes on the line is $d(s,t)/k$. □

Note that $d(s,t)/k$ may be different from $(c/(\alpha-1))^{1/\alpha}$. $(c/(\alpha-1))^{1/\alpha}$ is the theoretical value which minimizes (2) while $d(s,t)/k$ is optimal for integer number of hops. Therefore, the optimal number of hops and the optimal distance of adjacent nodes can be computed as long as locations of source and destination are provided.

3.2 Optimal Hop Count Routing

In this section, we propose a routing algorithm that is inspired by the optimal number of hops and space that are computed in section 3.1. The basic idea of our first algorithm is as follows. Each time, current node selects a neighbor node to progress, such that distance of the current node and the neighbor node is the closest to the optimal distance. That is, the neighbor that is the nearest to the optimal location is selected. Only neighbors closer to the destination than the current node are considered. Route failure will be reported to the source from the current node if there does not exist such a neighbor. The routing algorithm starts at source s. First, s rounds $d(s,t)\times((\alpha-1)/c)^{1/\alpha}$ to the nearest integer k, and compute the optimal distance of adjacent nodes $d(s,t)/k$. If $k\leq 1$ and $d(s,t)\leq r$, s transmits directly to t. Otherwise, s starts route discovery process. After that, each current node u on the route does the same as follows: Upon receiving a *Route Discovery* packet (except for s and t), if there is no neighbor closer to the destination than u itself, u reports *Route Failure* to s. Otherwise, u selects neighbor v such that $|d(u,v)-d(s,t)/k|$ is minimized. u passes route discovery task to v by transmitting v a *Route Discovery* packet attached with optimal distance $d(s,t)/k$. v is added into the route and continues route discovery. The route progresses hop by hop until destination t is reached or route failure is found. Detailed algorithm is formally presented as the following.

```
Optimal Hop Count Routing
Input: locations of $s$ and $t$.
Output: a route from $s$ to $t$.
Begin
round $d(s,t)\times((\alpha-1)/c)^{1/\alpha}$ to the nearest integer $k$;
compute optimal distance of adjacent nodes $d(s,t)/k$;
// for $s$.
if $k\leq 0$ and $d(s,t)\leq r$
  $s$ transmits directly to $t$;
while $t$ is not reached && Route Failure is not found do
  // for current node $u$ ($u\neq t$).
  if $\{v \mid d(u,v)\leq r, d(v,t)\leq d(u,t)\}=\Phi$
      $u$ reports Route Failure to $s$;
  else
      $u$ selects neighbor $v$ such that $|d(u,v)-d(s,t)/k|$ is minimized;
      $u$ transmits $v$ a Route Discovery packet attached with $d(s,t)/k$;
End
```

Once *t* is reached, the route from *s* to *t* is found. There are two strategies to move nodes on the route to their desirable locations. One strategy is to move in rounds as in [7]. Note that each node is assumed to know locations of its neighbors. In each round, each node (except for *s* and *t*) moves to the midpoint of its upstream node and downstream node on the route. Convergence of movements has been proven. However, it requires synchronous rounds in all nodes, and causes unnecessary zig-zig movement of nodes on the route. The other strategy is to move directly. Once *t* is reached, the number of actual hops is decided. *t* computes actual distance of adjacent nodes on the path, and attaches the information to a *Route Confirmation* packet that is routed backwards to all nodes on the route. Upon receiving a *Route Confirmation* packet (except for *s* and *t*), each node computes its desirable locations according to the attached actual space. After the node forwards the *Route Confirmation* packet towards *s*, it starts to move directly to its desirable location.

3.3 Minimum Power over Progress Routing

In this section, we propose a routing algorithm that minimizes transmission power of unit progress in selecting a forwarding neighbor. The basic idea of our second routing algorithm is to minimize transmission power over progress in each step. The routing algorithm starts at source *s*. Each time, current node *u* on the route does the same as follows: If there is no neighbor closer to the destination than *u* itself, *u* reports *Route Failure* to *s*. Otherwise, *u* selects neighbor node *v*, such that $(d(u,v)^{\alpha}+c)/(d(u,t)–d(v,t))$ is minimized. Note that $d(u,v)^{\alpha}+c$ is the transmission power for selecting *v* as a forwarding neighbor, and $d(u,t)–d(v,t)$ is the distance progress by node *v*. The metric denotes transmission power of unit progress. It is one of the cost over progress paradigms from [17]. *u* passes route discovery task to *v* by transmitting *v* a *Route Discovery* packet attached with location of *t*. Then *v* is added into the route and continues route discovery. The process continues until destination *t* is reached or route failure is found. Detailed algorithm is formally presented as the following.

```
Minimum Power over Progress Routing
Input: location of t.
Output: a route from s to t.
Begin
while t is not reached && Route Failure is not found do
  // for current node u (u≠t).
  if {v | d(u,v)≤r, d(v,t)≤d(u,t)}=Φ
        u reports Route Failure to s;
  else
        u selects neighbor v such that (d(u,v)^α+c)/(d(u,t)–d(v,t)) is minimized;
        u transmits v a Route Discovery packet attached with location of t;
End
```

Similar to the previous routing algorithm, move in round and move directly strategies are both adopted. Evaluation of these two strategies is showed in simulation.

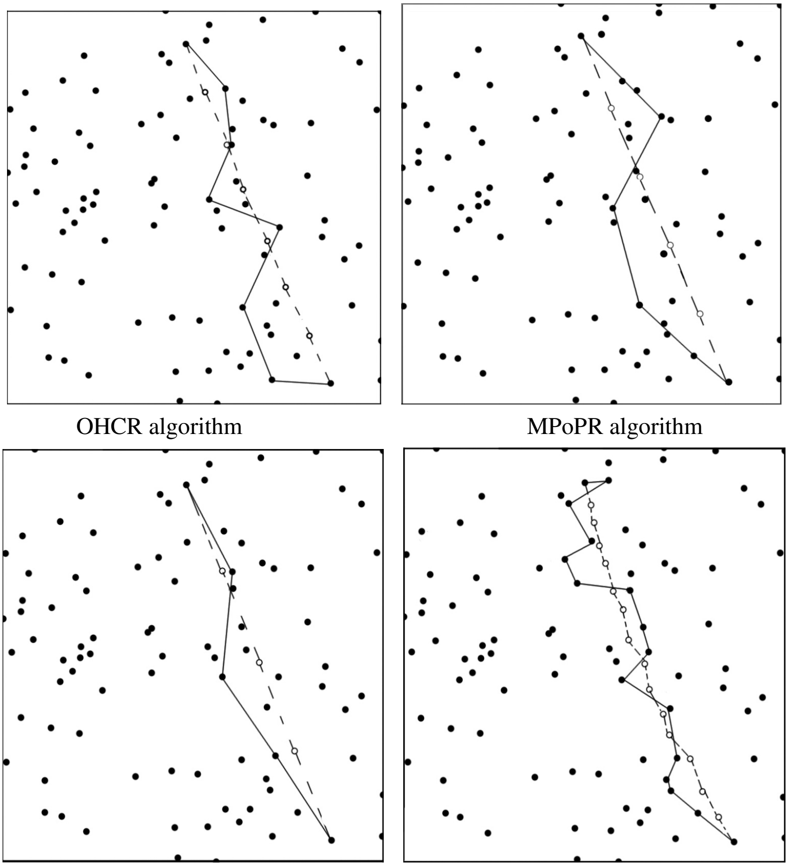
4 Simulation

MATLAB is adopted to evaluate performance of the proposed routing algorithms. In each simulation run, 250 nodes are randomly deployed on a square area with size varying from 50 to 50000. A source and a destination are randomly picked from the nodes. The network density D is defined as the average number of neighbors of each node. We set network density to 20 in our simulation. To control it, we adopt the method proposed in [13]. Suppose the network is a complete graph. All $(n-1)n/2$ edges are sorted into a list in non-decreasing order. We set transmission range r as length of the $(n \times D/2)$-th edge in the list. There is an edge between two nodes if and only if their Euclidean distance is not greater than r. Then, we obtain actual topology that meets required network density. We set such high density in our simulation to avoid frequent failure of route discovery. It is because only neighbors closer to the destination than current node are considered in our routing algorithms. Actually, in case there is no neighbor that is closer to the destination than current node, a recovery scheme such as face routing [3] can be applied. It is beyond the scope of the paper. Two power consumption models, HCB-model and RM-model [15], are both adopted in simulation. We set α=2 and c=2000 in HCB-model [14] while set α=4 and $c=2\times10^8$ in RM-model [13].

To analyze the performance of our proposed routing algorithms, we compare it with the routing algorithms in [7], where Greedy algorithm [6] and NP algorithm [18] are adopted to find an initial path before nodes move to desire locations in rounds. In greedy algorithm, current node on the route selects the neighbor that is closest to the destination while current node selects the neighbor that is closet to itself in NP algorithm. Only neighbors closer to the destination than current node are considered in both Greedy and NP algorithms. Otherwise, route failure will be reported. Four routing algorithms: Optimal Hop Count Routing (**OHCR** for short) algorithm, Minimum Power over Progress Routing (**MPoPR** for short) algorithm, Greedy algorithm, and NP algorithm are simulated and compared under the same environments. Move directly strategy and move in rounds strategy are both simulated. For the purpose of easily comparing, results of move in rounds and move directly are put together. We use “MD’ to denote move directly in following figures. The results presented in the following figures are the means of 100 separate runs.

4.1 Routing Paths Before and After Mobility Control

We first present an example to show how routing algorithms work before and after mobility control. 250 nodes are randomly placed in a 500×500 square. Source and destination are randomly selected. Four routing algorithms run on the same topology to find initial paths from the source to the destination. Then, move in rounds strategy is used to move nodes on the paths to their desired locations. The solid line in each figure is original path before mobility control, and the dashed line is the path after 20 rounds of mobility control. Hollow nodes in the dashed line are the nodes in the solid line after movement. Only a portion of the square is showed in Fig. 1 to see the paths clear. HCB model is adopted in the example, i.e., we set C=2000 and Alpha=2.



OHCR algorithm

MPoPR algorithm

Greedy algorithm

NP algorithm

Fig. 1. Routing paths before and after mobility control (*n*=250, *C*=2000, Alpha=2)

4.2 Decrease of Total Transmission Power

In this section, we simulate total transmission power of routes that are computed by four routing algorithms under HCB-model and RM-model. Both move in rounds and move directly strategies are considered. We study how total transmission power decreases step by step after mobility control in each round or direct moving.

From Fig. 2 and Fig. 3, we have following observations.

1. Both OHCR algorithm and MPoPR algorithm have better and more stable performance than Greedy algorithm and NP algorithm, especially in RM-model and the case of large squares.
2. OHCR has the best performance by using move directly strategy while MPoPR algorithm has the best performance by using move in rounds strategy. It is because that OHCR algorithm is based on optimal number of hops and space. Transmission power will dramatically decrease after direct moving.

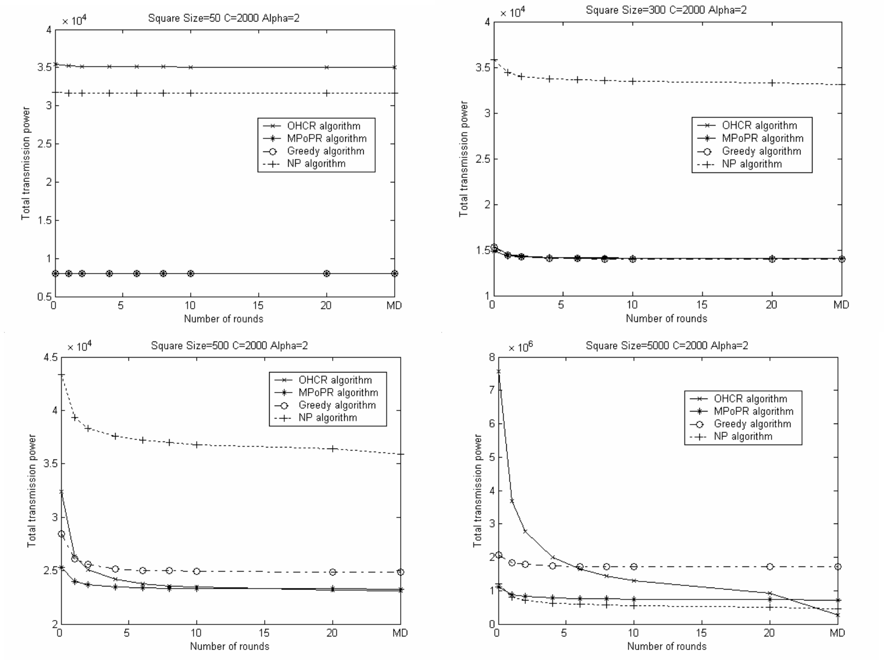


Fig. 2. Total transmission power under HCB-model

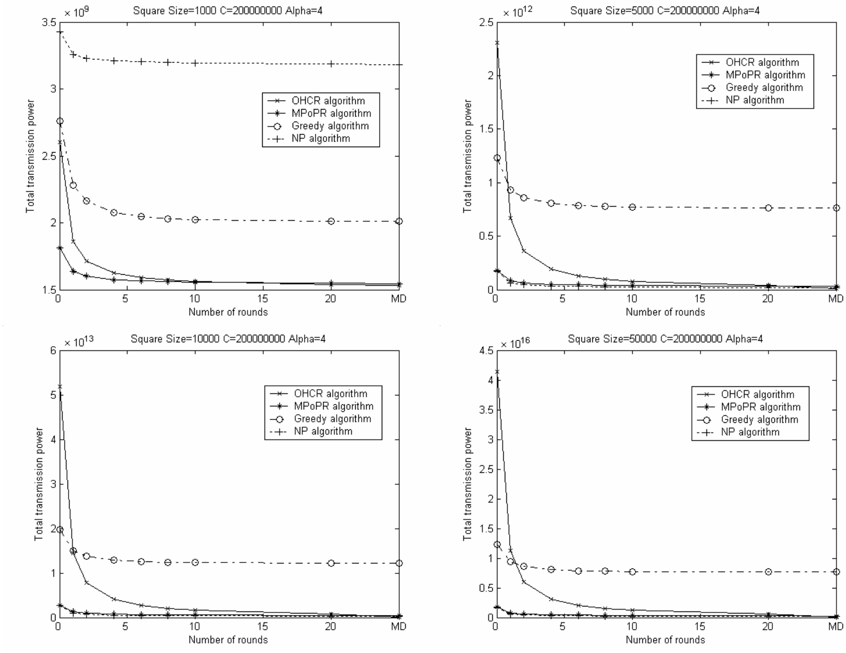


Fig. 3. Total transmission power under RM-model

3. Greedy algorithm performs well in only small squares while NP algorithm performs well in only large squares. Greedy algorithm selects the neighbor that is closest to destination. It results in less number of hops in the route. In small squares, transmission with less hops or direct transmission often save energy. In contrast, NP uses more hops that help to save energy in large squares.

4.3 Gain of Mobility

Note that mobility itself costs energy. Movement of nodes not only results in decrease of transmission power, but also causes extra energy cost on mobility. In this section, we study what is the gain of mobility control by integrating energy save on transmission power and extra energy cost on mobility. Once again, both move in rounds and move directly strategies are considered under HCB-model and RM-model. The gain of mobility is computed by $P_{initial}$–($P_{mobility\_control}$+P_M), where P_M is the extra energy cost on mobility in rounds or direct moving. Usually energy for mobility is proportional to distance traveled. For simplicity, the energy needed to move a node to distance M is assumed to be exactly M, i.e., P_M=M. In reality, it needs to be multiplied by some constant.

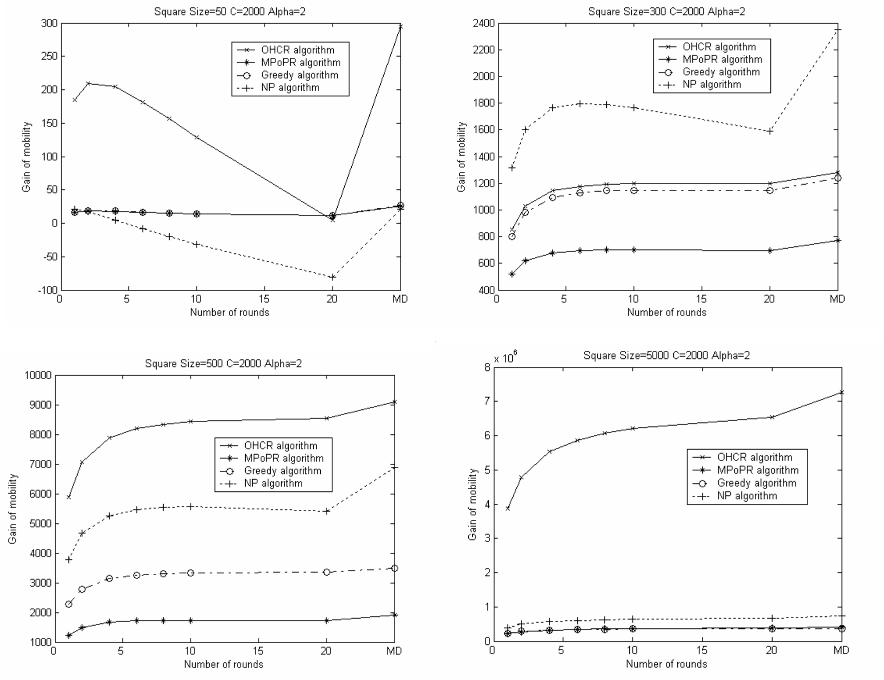


Fig. 4. Gain of mobility under HCB-model

From Fig. 4 and Fig. 5, we have following observations.

1. Gain of mobility is positive in most cases. Advantage of mobility control is more significant in bigger areas and under RM-model. It is because that transmission

power is larger in bigger areas and under RM-model. It provides more room for mobility control to reduce transmission power.

2. Gain of mobility increases as the number of rounds increases, and Move directly strategy is better than move in round strategy.
3. OHCR algorithm performs the best while NP algorithm performs the worst. It because that OHCR algorithm costs the least energy on mobility while Greedy algorithm costs the most energy in most cases.

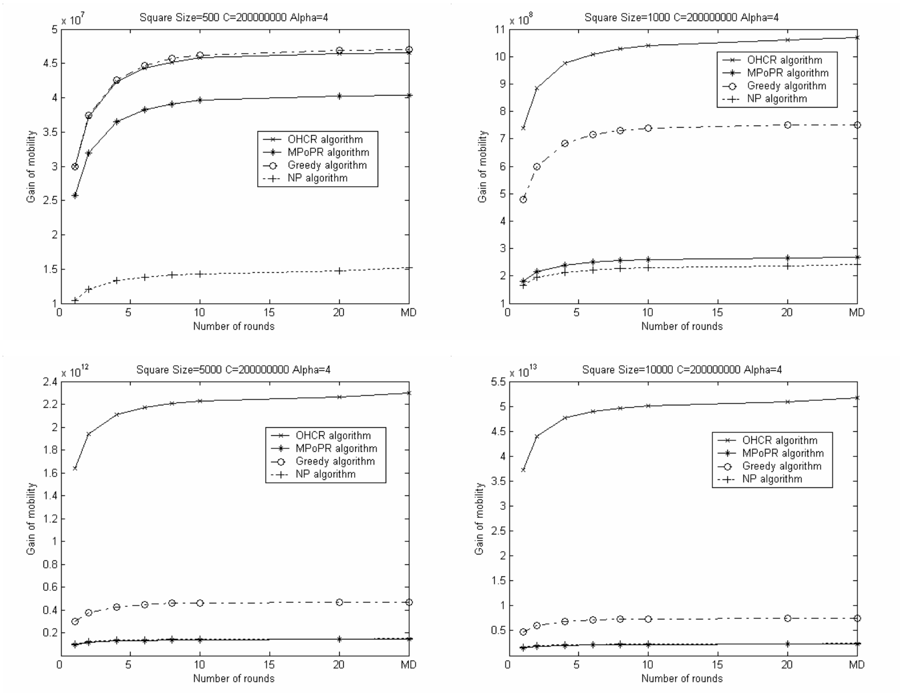


Fig. 5. Gain of mobility under RM-model

5 Conclusions and Future Work

The paper addressed mobility control routing problem in ad hoc and sensor wireless networks. Given a communication request between a source destination pair, we have studied the optimal number of hops and optimal distance of adjacent nodes on the route. We have proposed OHCR algorithm that integrates with the optimal number of hop counts, and MPoPR algorithm that minimizes transmission power over progress. Extensive simulations have been conducted to evaluate our proposed routing algorithms. According to simulation results in section 4, we have following are our main conclusions.

1. OHCR algorithm and MPoPR algorithm are more stable, more scalable, and use less transmission power than Greedy algorithm and NP algorithm.

2. Greedy algorithm performs well in only small squares while NP algorithm performs well in only large squares.
3. Move directly strategy costs less energy on mobility control than move in rounds strategy.
4. MPoPR algorithm has almost the best performance in case of without mobility control and moving within 20 rounds. OHCR algorithm becomes the best when move directly strategy is used.

There are some promising issues and problems need to be explored in future work. For example, an interesting problem is how to incorporate face routing into our framework for sparse networks. We focus on only power efficiency with mobility control in this paper. Actually, our framework can be applied to many aspects of networks. For example, mobility of nodes has been shown to be able to improve network capacity. One possible direction is to effectively use mobility control to improve network capacity.

Acknowledgements

This research is supported by NSERC Strategic Grant STPGP 336406, and the UK Royal Society Wolfson Research Merit Award.

References

1. Bachrach, J., Taylor, C.: Localization in sensor networks. In: Stojmenovic, I. (ed.) Handbook of Sensor Networks: Algorithms and Architectures, pp. 277–310. Wiley, Chichester (2005)
2. Bansal, N., Liu, Z.: Capacity, Delay and Mobility in Wireless Ad-Hoc Networks. In: Infocom 2003 (2003)
3. Bose, P., Morin, P., Stojmenovic, I., Urrutia, J.: Routing with Guaranteed Delivery in Ad Hoc Wireless Networks. ACM Wireless Networks 7(6), 609–616 (2001)
4. Dantu, K., Rahimi, H., Shah, H., Babel, S., Dhariwal, A., Sukhatme, G.: Robomote: Enabling Mobility in Sensor Networks. In: Proceedings of the 4th International Symposium on Information Processing in Sensor Networks, LA, California, vol. 55 (2005)
5. Dynia, M., Kutylowski, J., Lorek, P., auf der Heide, F.M.: Maintaining Communication Between an Explorer and a Base Station. In: Pan, Y., Rammig, F., Schmeck, H., Solar, M. (eds.) IFIP International Federation for Information Processing, vol. 216, pp. 137–146, Biologically Inspired Cooperative (2006)
6. Frey, H., Stojmenovic, I.: On Delivery Guarantees of Face and Combined Greedy-Face Routing Algorithms in Ad Hoc and Sensor Networks. In: ACM MOBICOM 2006, Los Angeles, pp. 390–401 (September 23-29, 2006)
7. Goldenberg, D.K., Lin, J., Morse, A.S.: Towards Mobility as a Network Control Primitive. In: Mobihoc 2004, Japan, pp. 163–174 (2004)
8. Grossglauser, M., Tse, D.N.C.: Mobility Increases the Capacity of Ad Hoc Wireless Networks. IEEE Transactions on Networking 10(4) (August 2002)
9. Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H.: Energy-efficient Communication Protocol for Wireless Microsensor Networks. In: Proceedings of the 33rd Annual Hawaii International Conference on System Sciences (January 2000)

10. Hofmann-Wellenhof, B., Lichtenegger, H., Collins, J.: Global Positioning System: Theory and Practice, 4th edn. Springer, Heidelberg (1997)
11. Howard, A., Mataric, M., Sukhatme, G.S.: Selfdeployment algorithm for mobile sensor networks. Autonomous Robots Special Issue on Intelligent Embedded Systems 13(2), 113–126 (2002)
12. Hu, D.L., Chan, B., Bush, J.W.M.: The hydrodynamics of water strider locomotion. Nature 427(7), 663–667 (2003)
13. Kuruvila, J., Nayak, A., Stojmenovic, I.: Progress Based Localized Power and Cost Aware Routing Algorithm for Ad Hoc and Sensor Wireless Networks. In: ADHOC-NOW 2004, Vancouver, Canada, pp. 294–299 (2004)
14. Liu, H., Jia, X., Wan, P.J.: On Energy Efficiency in Wireless Ad Hoc Networks. In: Xiao, Y., Pan, Y. (eds.) Ad Hoc and Sensor Networks, pp. 27–48. Nova Science Publishers (2005)
15. Luo, J., Hubaux, J.P.: Joint Mobility and Routing for Lifetime Elongation in Wireless Sensor Networks. IEEE Infocom 2005 (2005)
16. Rodoplu, V., Meng, T.H.: Minimum Energy Mobile Wireless Networks. IEEE Journal on Selected Areas in Communications 17(8), 1333–1344 (1999)
17. Stojmenovic, I.: Localized network layer protocols in sensor networks based on optimizing cost over progress ratio. IEEE Network 20(1), 21–27 (2006)
18. Stojmenovic, I., Lin, X.: Power Aware Localized Routing in Wireless Networks. IEEE Transactions on Parallel and Distributed Systems 12(11), 1122–1133 (2001)

A Hierarchical Multicast Routing Based on Inter-cluster Group Mesh for Mobile Ad Hoc Networks

Tomoyuki Ohta, Yasunori Kubo, and Yoshiaki Kakuda

Graduate School of Information Sciences, Hiroshima City University, Japan
{ohta@,yasu@pe.,kakdua@}ce.hiroshima-cu.ac.jp

Abstract. Multicast for ad hoc networks becomes popular. However, along with increase of the network size and node mobility, it is difficult to design the multicast routing for large ad hoc networks with high mobility. In order to cope with this difficulty, this paper proposes a new hierarchical multicast routing for such ad hoc networks. The characteristics of the proposed scheme are introduction of the inter-cluster group mesh in the autonomous clustering. The states of clusters are changed according to join and leave of multicast members and the group mesh among clusters is dynamically constructed. The simulation results show that the proposed scheme is scalable for the network size and adaptable to node mobility.

1 Introduction

An ad hoc network is a network consisting with mobile terminals and wireless links. The mobile terminals have the function of routers and they are called nodes hereinafter. In the ad hoc network, even if there are two nodes, which cannot directly communicate with each other via a wireless link, data can be indirectly delivered between them through connected wireless links. Since nodes move around an ad hoc network, the network topology frequently changes. Such the change has a big impact on the routing in the network.

With the spread of teleconference in the Internet, multimedia streaming such as music distribution, and peer to peer applications such as file sharing, demands of multicast communication increase for ad hoc networks. For the multicast communication, routes among members of a multicast group must be dynamically set up according to change of the network topology. MAODV [1,2] is a typical routing protocol for the multicast communication, in which a tree is formed for each group. However, MAODV has not the scalability. With increase of the network size, the tree maintenance becomes difficult due to frequent disconnection of wireless links. In addition, control packets for periodically confirmation of the group members and recovery of broken routes cause network congestion, which disturb efficient data delivery. In order to realize efficient multicast communication for large scale ad hoc networks, a new routing, which is able to adapt to change of the network topology and reduce the amount of control packets, is required.

This paper proposes a hierarchical multicast routing for ad hoc networks, which is based on inter-cluster group mesh in the autonomous clustering. The autonomous clustering is recognized in [3] as a kind of load balance clustering and referred as adaptive

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 32–43, 2007.

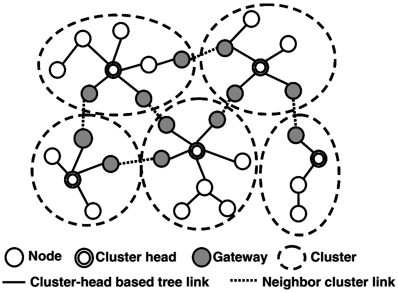


Fig. 1. Example of the autonomous clustering

multihop clustering. Since this clustering has load balancing capability, it is scalable for large ad hoc networks and adaptable to node movement. In the proposed multicast routing, a group mesh among clusters is dynamically constructed and data are delivered through mesh-like multiple routes among clusters. In the autonomous clustering, information on neighboring clusters is periodically updated. By adding information on multicast group to information on neighboring clusters, routes among clusters for multicast routing can be maintained with light load. The simulation results show that the proposed scheme is scalable for the network size and adaptable to node mobility in comparison with existing multicast routings.

The rest of this paper is organized as follows. Section 2 explains the autonomous clustering. Section 3 proposes a new multicast mesh routing. Section 4 gives simulation results for showing the effectiveness of the proposed protocol. Section 5 concludes this paper.

2 Autonomous Clustering Scheme

2.1 Definitions

An ad hoc network does not have mobile stations and wired links and is modeled by an undirected graph $G = (V, E)$ in which a node $v_i \in V$ represents a mobile host with ID i with switching capability and an edge $(v_i, v_j) \in E$ represents a wireless link between nodes i and j. When nodes v_i and v_j are connected by a link, they are said to be neighboring with each other. Since the nodes move around in the network, as time proceeds the links are connected and disconnected among the nodes and thus the network topology always changes.

2.2 Maintenance of Clusters

Figure 1 shows the example of the network formation which is constructed by the autonomous clustering. A cluster is a set of connected nodes which consists of a clusterhead and the other nodes. In the autonomous clustering scheme, the cluster is maintained so that the following properties are satisfied. ID of the clusterhead is ID of the cluster it manages and an arbitrary node in the cluster is connected by a sequence of wireless links between neighboring nodes whose ID are the same as that of the clusterhead. The size of the cluster is restricted by the lower bound L and upper bound U.

[Definition 1]: A cluster C whose clusterhead is v_i is called as C_i.
[Definition 2]: The number of nodes $|C_i|$ in the cluster C_i must be $L \leq |C_i| \leq U$.

Information each node has is a node ID, a cluster ID, a state and neighboring nodes. We describe the state in subsection 2.5 in detail.

The cluster ID of each node is autonomously changed by cluster ID's of the neighboring nodes. For example, if a node whose cluster ID is i is surrounded by all the neighboring node whose cluster ID is $j(\neq i)$, the cluster ID of the node is changed from i to j. The detail conditions for change of cluster ID's are given in [6].

2.3 Clusterhead

A node which manages the cluster is called clusterhead. The clusterhead forms a spanning tree in the cluster to efficiently collect information on all nodes in the cluster. The clusterhead periodically broadcasts ”clusterMEmber Packet” (MEP) in the cluster to form a spanning tree and inform that the clusterhead stays at the current cluster. If a node cannot receive the MEP from the clusterhead for a period, a new clusterhead must be selected in the cluster. Each node which receives the MEP and records the upstream node as the route toward the clusterhead in the routing table and broadcasts it to the downstream nodes. When each node receives ”clusterMember Acknowledge Packet” (MAP) from the downstream nodes, it records the downstream nodes as the route toward leaf nodes of the spanning tree in the cluster and sends MAP back to the upstream node in the route toward the clusterhead of the spanning tree.

Based on the collected information, the clusterhead makes a list of all nodes in the cluster. By collecting information on neighboring clusters, the clusterhead also makes a list of all neighboring clusters. Using these two lists, the clusterhead adjusts the number of nodes in the cluster as follows. When it is less than L, the clusterhead checks the sizes of all the neighboring clusters and merges the cluster with one of the neighboring clusters. When it is larger than U, the clusterhead divides the cluster to two clusters. Division and merger mechanisms for the autonomous clustering scheme are shown in [6,7] in detail. In either case, though information on neighboring clusters of merged or divided clusters is updated, influence of merger and division of clusters is restricted. Maintenance of clusters is thus locally performed.

2.4 Gateway

Node v_i is said to be a gateway if there is a neighboring node of v_i denoted by $v_j(j \neq i)$ whose ID is different from that of node v_i. The gateway can listen to MEP and MAP which are broadcasted in the different cluster. It can thus send MAP which contains the number of nodes and cluster ID in the different cluster to the clusterhead as information on the neighboring clusters. Owing to the MAP from gateways, the clusterhead can obtain ID's of neighboring clusters and ID's of gateways which connect to the neighboring cluster.

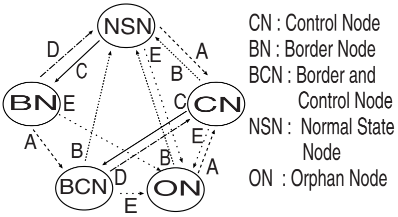


Fig. 2. State transition diagram of node v_i

2.5 States and Roles

In the autonomous clustering scheme, each node has a state and play a role corresponding to the state. The roles in the following states NSN, CN, BN, BCN, ON are explained below.

Normal State Node (NSN): In this state the node does not have any special functions for maintaining clustering such as clusterheads and gateways. Control Node (CN): In this state the node plays a function of the clusterhead. Border Node (BN): In this state the node plays a function of the gateway. Border Control Node (BCN): In this state the node plays functions of the clusterhead and gateway. Orphan Node (ON): In this state the node does not have any node ID. When a node joins the network, it is in this state.

2.6 State Transitions

Since even nodes which play essential roles for clustering such as clusterheads and gateways moves around in the ad hoc networks, whenever they cannot play them, instead some other nodes must play them. In order to maintain clustering even when nodes move, nodes autonomously change their states and they play roles corresponding to the states. The state of each node is changed according to change of states of the neighboring nodes. The state changes, in other words, state transitions are represented by Figure 2. The following five transitions A to E occur for maintenance of clustering.

A:v_i is selected as a new clusterhead. B:v_i moves to different cluster. C:v_i receives any control packets from a node whose cluster ID is different from v_i. D:v_i does not receive any control packets from nodes whose cluster ID's are different from v_i for a period. E:v_i does not receive any control packets from any nodes for a period. Conditions for the transitions are shown in [7,8] in detail.

Since nodes always move around in ad hoc networks, adaptability to node movement is one of the most important properties for clustering and routing.

3 Proposed Scheme

3.1 Outline

This paper proposes a new hierarchical multicast routing for ad hoc networks. The proposed scheme assumes that clusters are constructed in an ad hoc network by the

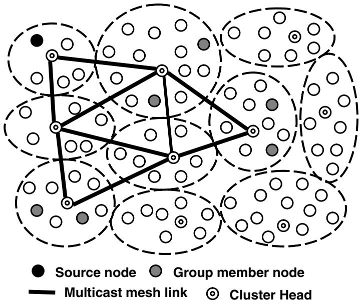


Fig. 3. Concept of hierarchical multicast routing based on intra-cluster group mesh

autonomous clustering. Different from the existing multicast routing that uses flooding among nodes, a hierarchical routing, in which each cluster is regarded as a virtual node and routes are set up among clusters, is adopted in the proposed scheme. In the proposed routing, routes among clusters for a specified multicast group are maintained by not a tree but a mesh (called group mesh, hereinafter) as shown in Figure 3. Reconfiguration of routes due to frequent break of routes can be thus avoided by alternative routes in the mesh.

In the proposed routing, a clusterhead in each cluster has a routing table for multicast routing, in which a pair of the cluster and the neighboring cluster (called mesh link, hereinafter) is recorded. Between every two neighboring clusters, a clusterhead in a cluster delivers data to the other clusterhead in the neighboring cluster through the mesh link. In order to avoid looping data delivery among clusters, if the clusterhead which has received data before receives the same data, it discards the data. In each cluster, data are delivered between the clusterhead and any node in the cluster through the clusterhead based tree formed in the cluster.

Request message and reply message for each group are used in the proposed scheme. A node which sends request messages for data is responsible for forming and maintaining the group mesh (such node is called core node) and the core node periodically request messages to all clusterheads in the ad hoc network.

3.2 Definition of Cluster States

Each cluster has the following four kinds of states for each multicast group. In oreder to maintain the group mesh, each clusterhead dynamically changes the cluster state whenever group member nodes move among clusters and mesh links among clusters are disconnected.

RC (Root Cluster): There is a core node in the cluster for maintaining the group mesh. MC (Multicast Member Cluster): There is a group member node in the cluster. FC (Forwarding Cluster): There do not exist any group member nodes in the cluster, however, there are two clusters whose states are RC or MC, which are connected by a sequence of mesh links on which the cluster is located, in other words, the cluster is located in the group mesh. NC (Normal Cluster): There do not exist group member nodes and the cluster is not located in the group mesh.

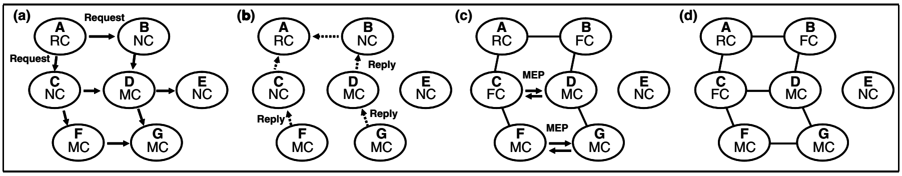


Fig. 4. Example of group mesh construction

3.3 Routing Table and Message Cache

Each clusterhead has the following two tables in the routing table.

(1)Reverse route table: This table includes source node of the request message and the next hop cluster on the route toward the cluster in which there is a core node. Whenever the clusterhead receives the request message, entries are added or updated. Whenever the clusterhead receives a reply message, it gives a next hop cluster from the table to which the clusterhead should forward the reply message. (2)Multicast Route Table: This table includes clusters which the clusterhead should send data for each group mesh and group member nodes for each group. Each entry is a cluster for each group which has a state of RC, MC, FC or NC.

Each node holds a message cache to detect duplication of packets Whenever each node receives a packet, the node stores from the packet the following: source node, destination node, message type and timestamp.

3.4 Join and Leave from Group

Information that a member node joins and leaves the multicast group is delivered to the clusterhead using control packets MAP's used in the autonomous clustering. When a node join the multicast group and becomes a member node, the clusterhead recognizes this fact by receiving MAP and checks the cluster state. If the state is MC, the number of group member nodes is incremented in the muticast route table. When a member leaves the group or a member node moves out of the cluster, the clusterhead recognizes this fact by receiving MAP and decrements the number of group members. If the number of group member nodes is zero, the clusterhead checks the number of mesh links to which the cluster is connected. If the number of mesh links is more than one, the clusterhead changes the state to FC. Otherwise, the clusterhead changes the state to NC and deletes the mesh link.

3.5 Construction of Group Mesh

An example of construction of the group mesh is shown in Figure 4. A node which first sends a request message for data packets becomes a core node of the group mesh and initiates construction of the group mesh. The core node sends a request message to the clusterhead through the clusterhead based tree in the cluster. The clusterhead forwards the request message to all the neighboring clusters and change the state to RC. When each clusterhead first receives a request message and stores it in the message cache, the clusterhead forwards it to all the neighboring clusters. When each clusterhead recognizes to receive the same request message by checking the message cache, the clusterhead discards it. Repeating these procedures, the request message is delivered to all clusters (see Figure 4(a)) and each clusterhead records the next hop cluster toward the cluster in which the core node exists in the reverse route table.

The clusterhead which received the request message checks the cluster state. If the state is MC or FC, the clusterhead sends a reply message toward the cluster whose state is RC (see Figure 4(b)). The clusterhead which sent the reply message to the neighboring cluster recognizes a pair of the cluster and the neighboring cluster as a mesh link. The reply message is delivered to all clusters in the group mesh whose states are FC, MC or RC. The clusterhead which received the reply message from the neighboring cluster recognizes a pair of the cluster and the neighboring cluster as a mesh link. The clusterhead checks the cluster state and if the state is NC, the clusterhead changes the state to FC after forwarding the reply message.

After exchanging of request and reply messages as described in the above procedures, routes among clusters whose states are FC, MC or RC are intermediately set up. After that, mesh links are newly added by control packets MEPs used in the autonomous clustering. As a result, mesh like routes among clusters whose states are FC, MC or RC are finally set up and construction of the group mesh has been completed (see Figure 4 (c), (d)).

3.6 Maintenance of Group Mesh

In order to reduce the load of the routing due to route recovery and reconfiguration, MEP's used in the autonomous clustering are used for maintenance of the group mesh. Information on the cluster state is delivered to the neighboring clusters through MEP.

When a clusterhead in each cluster (say cluster X) sends MEP in the autonomous clustering, if the cluster state is FC, MC or RC, in other words, cluster X is on the group mesh, the clusterhead add the cluster state to MEP. A gateway in the neighboring cluster (say cluster Y) receives such MEP and gets information on the state of cluster Y. The gateway forwards this information to the clusterhead in cluster Y through the clusterhead based tree by MAP. The clusterhead in cluster Y checks the state of cluster X and if the state is FC, MC or RC, in other words, cluster X is on the group mesh, the clusterhead in cluster Y checks whether there is a mesh link between clusters X and Y. If there do not such any mesh links, the clusterhead in cluster Y adds a mesh link in the multicast routing table.

When the state of a cluster (say cluster X) is changed to NC, in other words, the cluster X does not become on the group mesh, this information is delivered to another clusterhead in the neighboring cluster (say cluster Y) using MEP in cluster X and MAP in cluster Y in a similar way as described in the above procedure. The clusterhead in cluster Y checks whether there is a mesh link. If there is such a link between X and Y, the clusterhead deletes the mesh link in the multicast routing table.

Addition and deletion of mesh links using MEP in the autonomous clustering are triggered by the following cases. (a) Case that a new cluster emerges: When a new clusterhead receives MAP which includes a member node, the clusterhead changes the state to MC and send MEP which includes the changed state. When a new clusterhead receives MAP including the state of neighboring cluster, which is FC, MC or RC, the clusterhead changes the state to FC, MC or RC. (b) Case that a clusterhead finds a new neighboring cluster: When a clusterhead receives MAP indicating that a new neighboring cluster includes a member node, the clusterhead adds a mesh link to the neighboring cluster. (c) Case that a reply message is lost: Due to change of the clusterhead based tree or congestion, there are possibilities that a reply message is lost. If a pair of states of two neighboring clusters between which a reply message is lost is (RC,MC), (RC,FC), (MC,MC), (MC,FC), (FC,FC), a mesh link is set up by exchanging MEP between such clusters.

Each clusterhead periodically sends MEP to all nodes in the cluster and all gateways in the neighboring clusters. As a result, each clusterhead can receive MAP which includes neighboring clusters and record them. Each clusterhead checks difference between the current neighboring clusters and the recorded neighboring clusters. If the destination of a mesh link is in the recorded neighboring clusters, however, disappears in the current neighboring clusters, the mesh link is recognized to be disconnected and deleted. After deletion of the mesh link, if the state of the cluster is FC and there is only one mesh link from the cluster, the mesh link is also deleted and the state of cluster is changed to NC, in other words, the cluster leaves from the group mesh.

4 Simulation Evaluation

4.1 Simulation Environments

In order to show the effectiveness of the proposed hierarchical multicast routing based on inter-cluster group mesh, the simulation experiments have been performed in comparison with MAODV. In addition, we have evaluated the proposed scheme and the proposed scheme which does not utilize the inter-cluster group mesh but inter-cluster tree (shortly, no mesh scheme) to verify the effect of the inter-cluster group mesh. In the no mesh scheme, the multicast tree among clusters, which are constructed and maintained by the autonomous clustering scheme, is constructed only using Request and Reply messages.

We used QualNet 3.9 [4] as the network simulator. Tables 1 and 2 show the simulation environments and the parameters on the multicast group, respectively. In addition, Tables 3 and 4 show the parameters used for the proposed scheme and MAODV, respectively.

As shown in Table 1, we adopt the random waypoint model [5] as the node mobility model and set the pause time at 0. In the simulation experiments, we adopt MCBR as the application working on each node. MCBR is CBR (Constant-Bit-Rate) for multicast application, whose destination address is set at the multicast address. After 60 seconds from the simulation start, each source node starts to send data packets at 0.25 seconds interval to the group members in the multicast group. The field sizes are 1500m × 1500m, 2800m × 2800m, 3500m × 3500m, 4300m × 4300m, and 5000m × 5000m, respectively, in case that the numbers of nodes are 100, 300, 500, 700, and 1000. In all the graphs shown in this section, we denote the proposed scheme, the no mesh scheme, and MAODV as ’proposal’, ’proposal (no mesh)’, and ’MAODV’, respectively.

Table 1. Simulation Environments

Field size (m^2)	1500 x 1500, 2800 x 2800, 3500 x 3500, 4300 x 4300, 5000 x 5000
Number of nodes	100, 300, 500, 700, 1000
Maximum node speed (m/s)	1, 5, 10
Node mobility	Random waypoint model
Simulation time (s)	320
Pause time (s)	0
MAC layer	IEEE 802.11
RTS / CTS	off
Transmission range (m)	250
Bandwidth ($Mbyte$)	54

Table 2. Parameters used for Multicast Group

Number of multicast groups	5
Number of multicast members	10
Interval of sending MCBR packet (s)	0.25
MCBR packet size ($byte$)	512
Max number of delivered data packets	50000

4.2 Simulation Plan

We have evaluated the proposed scheme in comparison with MAODV with respect to the total control packet size and the number of delivered data packets. As shown in Table 2, five multicast groups are constituted in the network and each multicast group consists of one source node and ten member nodes. In order to measure the results shown in graphs and tables, in each case, the simulation experiment is run five times and the average value is calculated by the five simulation results.

4.3 Results for Total Control Packet Size

Figure 5 shows the total control packet size versus the number of nodes when the node speeds are 1, 5, and 10 m/s, respectively. In case that the node speed is 1 m/s and the number of nodes is small, the total control packet size in MAODV becomes smaller than that in the proposed schemes. In this case, since the topology does not change frequently, the number of route breaks on the multicast tree which is constructed by MAODV becomes lower, and thus the route repair does not occur frequently. However,

Table 3. Parameters used for the proposed scheme

Cluster size (L,U)	(20, 50)
Interval of clustermember packet (s)	2
Interval of request broadcasts (s)	5

Table 4. Parameters used for MAODV

Number of allowed hello losses	3
Interval of group hello broadcasts (s)	5
Interval of hello broadcasts (s)	1

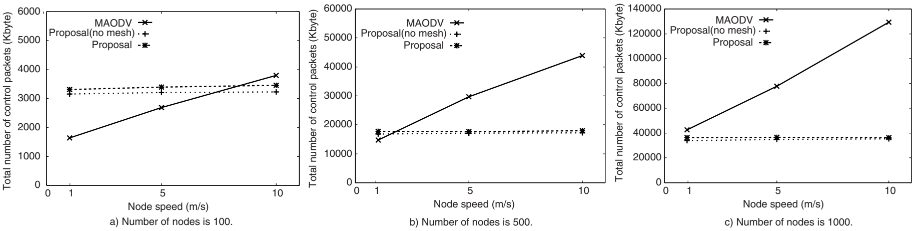
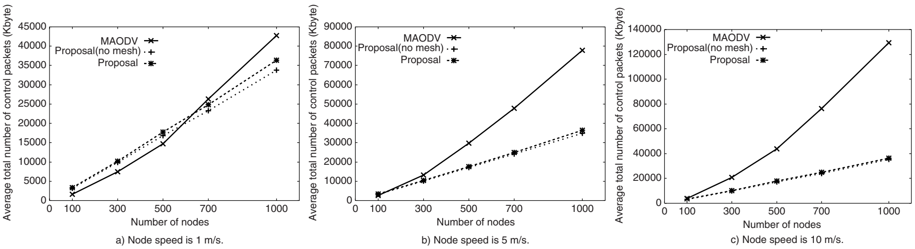


Fig. 5. Total control packet size versus the node speed when the numbers of nodes are 100, 500, and 1000

as the node speed becomes faster, the total control packet size in MAODV increases. This is because the number of route breaks on the multicast tree increases, and thus many control packets to repair the tree are used.

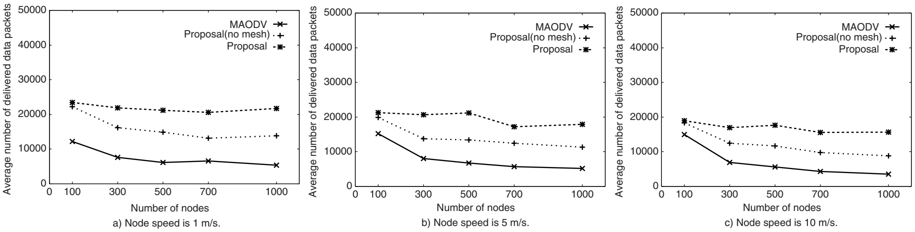
On the other hand, the total control packet size in the proposed scheme is almost the same regardless of the node speed. However, as shown in Figure 5, in case that the node speed is 1 m/s, the total control packet size in the proposed scheme becomes larger than that in MAODV when the numbers of nodes are 100 and 500. In the proposed scheme, the clusterhead in each cluster periodically broadcasts MEP's in the cluster and the clustermembers send MAP's back to the clusterhead to maintain the cluster. Most of control packets in the proposed scheme are used to maintain the cluster, and the number of control packets increases as the number of nodes becomes larger. Therefore, since the proposed scheme has the overhead to maintain the cluster without depending on the number of route breaks, the proposed scheme has more overhead than MAODV when the node speed is low. On the contrary, the total control packet in the proposed scheme is much less than that in MAODV when the node speed is 10 m/s. In addition, the total control packet size does not increase and decrease even if the node speed becomes faster. It means that the control packet used for the proposed scheme does not mainly consist of the control packet used for maintaining the group mesh but the control packets used for the autonomous clustering. As a result, we can say that the overhead of the proposed scheme becomes relatively constant regardless of the node speed.

Figure 6 shows the total control packet size versus the number of nodes when the node speed are 1, 5, and 10 m/s. As shown in Figure 6, as the number of nodes becomes larger, the total control packet size in MAODV becomes much larger than that in the proposed scheme. The total control packet size MAODV increases in proportional to the increase of the number of nodes. It is because the route break frequently occurs when the numbers of nodes are 700 and 1000, and thus the large number of control packets is generated to repair the multicast tree. Therefore, as the node speed becomes faster, the route break occurs more frequently, and thus the overhead of MAODV becomes much higher.



a) Node speed is 1 m/s.
b) Node speed is 5 m/s.
c) Node speed is 10 m/s.

Fig. 6. Total control packet size versus the number of nodes when the node speeds are 1, 5, and 10 m/s



a) Node speed is 1 m/s.
b) Node speed is 5 m/s.
c) Node speed is 10 m/s.

Fig. 7. Number of delivered data packets versus the number of nodes when the node speeds are 1, 5, and 10 m/s

On the other hand, the total control packet size in the proposed scheme is almost the same regardless of the node speed. The proposed scheme can be adaptive to the topology change and thus the cluster structure does not often change thanks to the autonomous clustering scheme. Since the multicast group mesh is constructed by regarding one cluster as one virtual node, the route between the virtual nodes (clusters) does not break frequently. Therefore, it means that the control packets used for maintaining the group mesh are not generated even when the node speed is high. Especially, in case that the number of nodes is 1000 and the node speed is 10 m/s, the total control packet size in the proposed scheme becomes one third of that in MAODV. Therefore, we can say that the proposed scheme works efficiently since it mitigates the overhead even when the node speed is high.

4.4 Results for Delivered Data Packets

Figure 7 shows the number of delivered data packets versus the number of nodes when the node speeds are 1, 5, and 10 m/s. The number of delivered data packets in the proposed scheme much more than that in MAODV regardless of the node speed. Especially, when the number of nodes is 1000, the number of delivered data packets in the proposed scheme is four times as many as that in MAODV. It is expected that in MAODV the network congestion occurs since a large number of control packets are generated for repairing the tree due to the topology change as shown in Figure 6. As a result, in MAODV, many data packets are lost due to the network congestion.

On the contrary, the proposed scheme provides the multiple routes from the source node to each multicast group member to construct the inter-cluster group mesh. Since the number of data packet forwardings in the proposed scheme increases more than that in the no mesh scheme, each multicast group member can receive more data packets.

As a result, we can say that the proposed scheme becomes more effective than MAODV, especially when the number of nodes becomes larger.

5 Conclusion

This paper proposed a new hierarchical multicast routing for ad hoc networks. The proposed scheme is based on the inter-cluster group mesh and the autonomous clustering. The proposed scheme has been compared with a typical multicast routing MAODV and the scheme based on the inter-cluster tree and the autonomous clustering by simulation experiments. The simulation results shows that the proposed scheme is more effective than these two schemes from the viewpoints of the number of control packets and the number of delivered data packets in cases that the network size is large and the node mobility is high. We conclude that the proposed scheme is therefore scalable and adaptable for multicast routing in ad hoc networks.

Acknowledgments

This work was supported in part by the Ministry of Internal Affairs and Communications of Japan under Grant-in-Aid for SCOPE-C (No.052308002), the Ministry of Education, Science, Sports and Culture of Japan under Grant-in-Aid for Scientific Research (C) (No.18500062) and Young Scientists (B) (No.18700070), and Hiroshima City University under their research grants.

References

1. Royer, E.M., Perkins, C.E.: Multicast operation of the ad-hoc on-demand distance vector routing protocol. In: MOBICOM. Proc. ACM/IEEE International Conference on Mobile Computing and Networking, pp. 207–218 (1999)
2. Zhu, Y., Kunz, T.: Maodv implementation for ns-2.26. Systems and Computing Engineering, Carleton University, Technical Report SCE-04-01 (2004)
3. Yu, J.Y., Chong, P.H.J.: A survey of clustering schemes for mobile ad hoc networks. IEEE Communications Surveys & Tutorials 7(1), 32–48 (2005)
4. QualNet Network Simulator by Scalable Network Technologies, Project web page available at http://www.scalable-networks.com/
5. Broch, J., et al.: A performance comparison of multi-hop wireless ad hoc network routing protocols. In: Proc. ACM/IEEE MOBICOM 1998, pp. 85–97 (1998)
6. Ohta, T., et al.: An improved autonomous clustering scheme for highly mobile large ad hoc networks. In: Proc. 7th IEEE International Symposium on Autonomous Decentralized Systems (ISADS2005), Workshops on AHSP2005 1st International Workshop on Ad Hoc, Sensor and P2P Networks, pp. 655–660 (2005)
7. Ohta, T., et al.: An adaptive multihop clustering scheme for highly mobile ad hoc networks. In: ISADS2003. Proc. 6th IEEE International Symposium on Autonomous Decentralized Systems, pp. 293–300 (2003)
8. Ohta, T., et al.: An adaptive multihop clustering scheme for ad hoc networks with high mobility. IEICE Trans. Fundamentals E86-A(7), 1689–1697 (2003)

Load-Balancing Using Multi-path Directed Diffusion in Wireless Sensor Networks[*]

Arash Nasiri Eghbali and Mehdi Dehghan

Ad Hoc Network Laboratory, Computer Engineering Department
Amirkabir University of Technology, Tehran, Iran
{a.eghbali, dehghan}@aut.ac.ir

Abstract. Directed diffusion (DD) is a data-centric routing protocol based on purely local interactions between individual network nodes. This protocol uses application-specific context for data aggregation and dissemination. Therefore, it can be completely matched to the application requirements in a large distributed sensor network. Many work have been recently done to improve the energy efficiency of this protocol. In this paper, an extension to DD is presented in order to construct multiple paths between the sink and the sensor sources. Using this method, load-balancing is implemented to increase the life-time of the sensor nodes collaborating in the routing process. The proposed protocol, Multi-path directed diffusion (MDD), can produce more than one disjoint or braided paths and spread the data collected in the sources, properly between the paths. In this way, an efficient load balancing mechanism has been implemented. The simulation results show that through using MDD, the lifetime of the network connections between the sources and the sink will be increased and the interest flooding rate which is proved to be an expensive operation can be reduced.

Keywords: wireless sensor networks; multi-path routing; load balancing; energy efficiency

1 Introduction

In recent years, we have witnessed a growing interest in the field of wireless sensor networks (WSNs) which consist of a large number of micro-sensor nodes distributed in a large environment to gather and process specific information required for different applications. In such networks, individual nodes are not particularly important and usually they have not a unique address.

Directed diffusion (DD) [4] is a data-centric routing protocol proposed for data gathering in wireless sensor networks. In DD, attribute-value pairs are used for describing the information. This algorithm in its basic form has two phases. In the first phase, the sink node floods a request packet called "interest" containing the desired attribute-value pairs. When this packet reaches a source node that has the requested information (second phase), the source node floods an "exploratory data"

[*] This work was supported by Iran Telecommunication Research Center (ITRC).

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 44–55, 2007.

(ED) packet through the network. When the ED packet reaches the sink node, it will send a “positive reinforcement” packet toward the source node. This packet is being forwarded through the path traversed by the ED packet. In this way, a bi-directional path is constructed between the sink and source nodes. Afterwards data packets will be sent through the reinforced path. This algorithm is also called two-phase-pull (TPP) algorithm. By assuming the connections to be bi-directional (which in most cases is not true) One-Phase-Pull (OPP) algorithm can be used. In this approach, data packets are sent immediately by the source node after receiving the interest packet. Therefore, in OOP algorithm the flooding overhead of ED packets will be omitted.

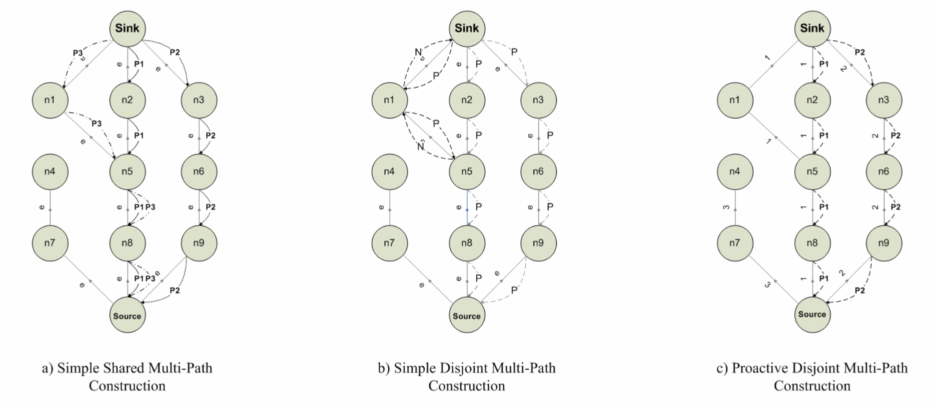
The design of mechanisms for single-path routing in sensor networks has been explored in [1]. To provide connection between the sink and the sources, this work relies on low-rate flooding of events that enabled local re-routing when the nodes in the primary path have failed due to energy consumption. In sensor networks, where energy efficiency is of paramount importance, such flooding can adversely impact the lifetime of the network. To provide better connection time and implement load-balancing between the sink and the source, it is desirable to find alternative techniques.

The other problem of this approach is the mechanism used for routing selection which mostly leads to select the shortest path between the sink and the sources. In this case the nodes in the shortest path will fail after a short period of time due to lack of energy. This problem is intensified when the nearest route to the shortest path is selected after refresh period which is quite probable. In this situation, network partitioning will occur along the depleted paths.

We propose using multipath routing algorithms to increase load-balancing between network nodes during forwarding data packets between the sink and the sources. In this way, we can reduce the rate of interest packet flooding by increasing the lifetime of connection. Multipath routing techniques have been discussed in literature for several years now. However, using of multipath routing algorithms in sensor networks in order to gain energy efficiency and load-balancing has not yet been explored. In sensor networks, designing such algorithms has been proved to be a difficult task, due to the data-centric routing with localized path setup [5].

We consider two different approaches to construct multiple paths between the sink and the source: *node-disjoint multi-paths* where the alternate paths do not intersect each other. The disjoint property has better performance for load-balancing but reduces the number of paths created by the algorithm. Also if a single node in a disjoint path fails, other nodes will be left unused until the next refresh period. The other approach abandons the disjoint requirement and builds *braided paths*. Our definition of braided paths is the same as [5] but we used multipath routing for load-balancing and the work in [5] is focused only on the resiliency in the presence of failure.

In this paper we introduce the *premier packet problem* where the first flooded packet dominates most of the nodes between the source and the sink and prevents constructing multiple paths. We propose four methods to improve the probability of shaping multiple paths. At first we present the random forward improvement (*RFI*) method. In this method, the probability P is considered for a node whether to broadcast an exploratory data packet or not. By reducing this probability, other packets would have a better chance to reach the destination. Another improvement is using a random delay before broadcasting each exploratory data. The third method is called limited forward improvement (*LFI*) in which each node would selectively send the exploratory data to the nodes being nearer to the destination instead of broadcasting them. In this way, we can achieve a significant reduction in the overhead imposed by [1] to broadcast the exploratory data packets all through the network. The number of constructed paths would also grow using this improvement method.



a) Simple Shared Multi-Path Construction

b) Simple Disjoint Multi-Path Construction

c) Proactive Disjoint Multi-Path Construction

Fig. 1. Comparison between Simple and Proactive Multi-Path Construction Algorithms

2 Multi-path Creation Methods

In this section, we introduce three different methods for multi-path creation in the MDD routing protocol and in the next section we would have a brief evaluation and comparison of these methods.

A. Simple Multi-Path Directed Diffusion Method (S-MDD)

Unlike DD in which the sink node only reinforces the first arriving exploratory data packet and simply drops the others, in this method the sink node reinforces all of the received exploratory data packets. Also when an old *positive reinforcement* packet arrives in a node in the path (*path node*), instead of just dropping the old reinforcement packet (as in DD), it will send a *negative reinforcement* message to its previous hop. This is done in order to avoid forming none-disjoint paths between the sink and the sources. S-MDD is used in [5] to increase the resiliency of directed diffusion routing protocol.

As another approach, *path nodes* can also forward the old positive reinforcement message, through the path already traversed by the prior reinforcement message. In this way, the paths may not be completely disjoint but the number of constructed paths will be fairly increased. These methods are explained in figure 1. In this figure, node S is the source node and node D represents the sink. The exploratory data (ED) that firstly arrived in each node in phase II of two-phase-pull (TPP) version of DD are labeled as ‘e’. The shared path creation procedure is shown in the figure 1.a. In this case, node n5 forwards the packet through the path, traversed by the first positive reinforcement message P1. In figure 1.b we have depicted the disjoint path creation method. As it is shown, in this method node n5 sends a negative reinforcement message to node n1 after receiving the old P3 reinforcement message.

B. Disjoint Proactive Multi-Path Directed Diffusion Method (DP-MDD)

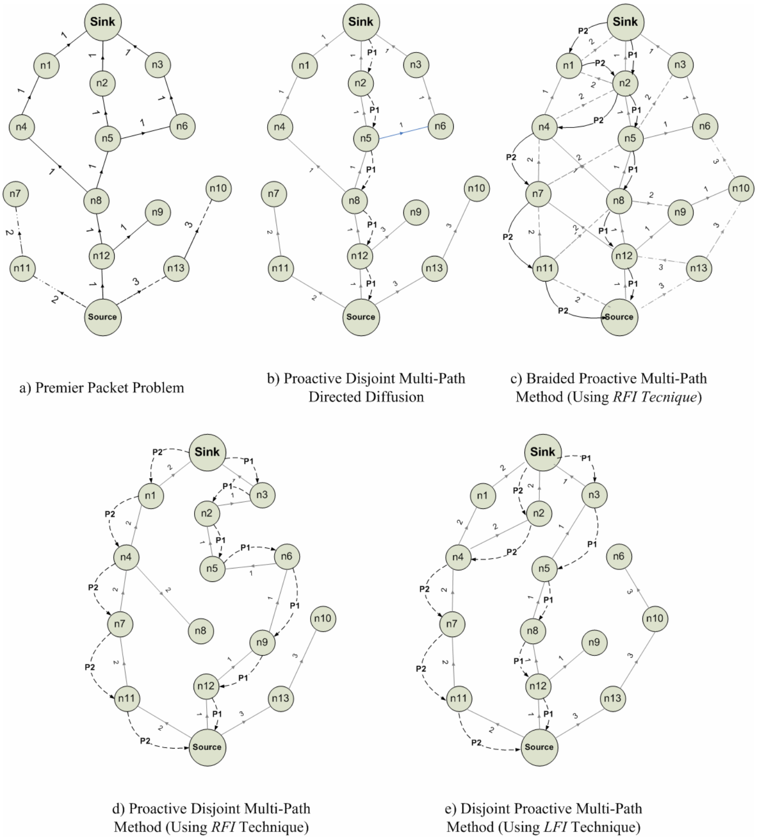
DP-MDD method produces multiple paths proactively by tagging the ED messages in the source node, before forwarding them to the neighbor nodes. We call this tag, multi-path identifier as *MP_ID*. Basically, this approach acts similarly as the simple disjoint path creation method but when an exploratory data arrives in the sink node, it will be recognized as a new exploratory data (ED) with a new MP_ID or a duplicated one. So the sink reinforces only the first ED packet among those arriving with the same MP_IDs. In this way, we will be assured that created paths are disjoint (the old ED packets are ignored in the base DD algorithm).

This method is shown in the figure 1.c. in which the sink node receives the ED packets with MP_IDs 1 and 2. Firstly the sink receives the ED from node n2 with MP_ID, then it receives the ED from node n3 and finally receives the ED from n1. So the sink reinforces only the gradients from node n2 and n3 and ignores the ED received from n1.

The DP-MDD and S-MDD algorithms are based on the arrival of ED packets in the sink node, with totally distinct *path nodes*. Unfortunately this rarely happens because of the broadcasting and physical nature of wireless medium, which reduces the probability of arriving multiple different ED packets. The first packet arrives at the sink node, usually traverses all the neighbor nodes and the other ED packets can hardly reach them. This phenomenon will be explained in the next section.

Four improving rules are suggested to increase the number of the disjoint paths:

- *Random Forward Improvement Method (RFI):* In this method, each node sends the ED packets to each neighbor with a *P* probability during broadcasting of ED packets (Figure 2.d).
- *Random Delay Improvement Method (RDI*): In this method each node waits for a random period of time *D*, before sending the ED packets to each neighbor node.
- *Limited Forward Improvement method (LFI):* In this method, each node selectively forwards the ED packets to first *F* nearest nodes to the sink. The distance is measured by the time-stamp, saved during the interest message broadcasting (in the first phase of the TPP algorithm). An example is shown in figure 2.e. Another selecting approach is using the P parameter to forward the packet to first neighbor. Then after each forward, *P* will be reduced exponentially. Another way for limiting the selection is using a delay before sending each ED packet to each neighbor. Then after each forward, the delay period will be increased exponentially. We should use this method for first *F* packets unless some of the packets will be sent after the time out period and they will be simply discarded.
- *Hybrid Improvement Method (HI):* in this method, we combine the previous methods and each node, forwards the ED packet with a random delay *D* and the probability *P* with the forwarding factor *F*.



a) Premier Packet Problem

b) Proactive Disjoint Multi-Path Directed Diffusion

c) Braided Proactive Multi-Path Method (Using *RFI Tecnique*)

d) Proactive Disjoint Multi-Path Method (Using *RFI* Technique)

e) Disjoint Proactive Multi-Path Method (Using *LFI* Technique)

Fig. 2. Comparison between Simple and Proactive Multi-Path Construction Algorithms

C. Braided Proactive Multi-Path Directed Diffusion (BP-MDD)

Here, each node can forward more than just a single ED packet. In this case, the duplicated ED packets with different MP_IDs are not discarded and each node is permitted to forward at most *T* duplicate packets. We call this parameter, the *node forwarding threshold* parameter. Therefore, the paths constructed are not necessarily disjoint and can share a number of nodes. Yet we ensure that the paths are braided because the gradients are not shared between two different paths and each gradient can only be a member of a single path. This rule prevents the paths to share more than a few nodes. By using the parameter *T*, more paths will be produced. Therefore when a single node in a path runs out of energy, other path nodes have a better chance to be engaged in the routing process.

This technique mostly reduces to DP-MDD algorithm when each node broadcasts the ED packet, because the packets are broadcasted to all neighbors, at the same time. Therefore all the gradients from neighbors will be tagged with the MP_ID of the first packet, broadcasted. So the paths constructed with this algorithm, without implementing the improvement techniques are usually disjoint. To form braided paths, the RFI or the LFI method should be used. Although the number of ED packets forwarded, do not increase by the order of *T* (because of omitting shared gradients), the protocol overhead is hardly negligible without using LFI or RFI methods. A sample of this protocol has been figured in the figure 2.c. in this figure we assumed the *T* parameter to be 2 and LFI improvement has been implemented.

3 Evaluation and Comparison

In this section, a brief review about the behavior of multi-path construction methods, presented in section 2, will be given. Also we will make a comparison between these methods and summarize our main choices for multiple paths producing algorithms. In section 4, we will compare our evaluations with the simulation results.

The functionality of simple multi-path creation methods mostly is related to the topology of the network. The main weakness of this method relies upon this fact that it rarely can produce multiple disjoint paths. Disjoint routes, starting from the source node, will merge and become a single path after traversing just a few nodes. The main reason behind this problem is the nature of wireless medium and contentions occur during flooding a packet in a wireless network. When the sink node broadcasts an ED packet to its neighbors and they forward the packets to their neighbors, the wireless medium will soon become saturated. In this case the first ED packet that is sent towards the source node (out of the saturated area), can traverse most of the nodes between the source and the sink nodes. As we mentioned in the previous section, we call this event the “premier packet problem”. The directed diffusion algorithm makes use of this event for data aggregation but this event prevents the simple and proactive routing methods to make multiple disjoint paths. This event has been depicted in figure 2.a. In this figure the ED packet with MP_ID 1 has reached the sink before others.

The S-MDD (Simple) algorithm in its shared form can produce a number of different paths (by the size of all neighbors of the source node), but as we mentioned in the above paragraph, the paths will merge after a few hops and therefore, they are not usable for our load-balancing purpose. The DP-MDD in its base form and the S-MDD algorithm while creating disjoint paths, behave similarly. As it has shown in figure 2.b, these algorithms cannot create more than one single path in the situations that the premier packet has occupied most of the nodes between sources and sinks.

By Using *RFI* (Random Forward) improvement method for disjoint proactive algorithm, ED packets with the same MP_IDs, would have a little chance to dominate all the nodes between the source and the sink and the effect of *premier packet* will be reduced. This effect will be increased by decreasing the value P. The main problem with this improvement method is its random behavior. So the paths constructed will be not straight and this increases the mean length of produced paths (in this paper we consider the hop count as the length of a path). This distortion will be increased with decreasing the value *P* and also our experience shows that an ED packet would have a little chance to even reach the destination with little *P* values. As we know, the energy consumed for routing process directly increased by the length of the path. So we would like to create paths with the minimum possible length.

By using *RDI* (Random Delay) method, we would have more straight paths but the number of disjoint paths will be decreased. Unfortunately this method usually provides poor results and sometimes worst than the unimproved version of the algorithm. Another problem with this method is the latency imposed to ED packet while flooding, and this could cause problems for delay sensitive applications, especially in large scaled sensor networks.

The *LFI* (Limited Forward) improvement method has better results than prior methods and can usually create a notable number of disjoint paths. Another advantage of using this method is reducing the cost of flooding ED packets. Because of its limited forwarding, the number of flooded ED packets will be increased linearly by the size or density of the network. Although in this method, the nodes should send the ED packets to each neighbor instead of broadcasting this method can reduce the total number of communication overhead.

The Limited forwarding method stands somewhere between the two proposed one-phase-pull (OPP) and two-phase-pull (TPP) algorithms. In the OPP method, the ED packets are omitted and the source node will send data packets to the first neighbor node that has sent the interest packet. In this algorithm we assume the connections between nodes to be bi-directional but this assumption usually is not true. The TPP algorithm has also the problem of large overhead, caused by flooding ED packets through the whole network. In the MDD algorithm, the connections need not to be bi-directional and when using *LFI* improvement techniques, the overhead of ED packets are almost negligible comparing to the TPP algorithm.

The BD-MDD (Braided Proactive) method can increase the number of paths between source and sink but these paths relax the requirement for node disjointedness. Alternate paths in a braid are partially disjoint from each other, not completely node-disjoint. Figure 2.c illustrates a braid obtained by the BD-MDD algorithm. The *RFI* and the *LFI* improvement methods can improve the performance of this algorithm, significantly.

4 Energy-Efficiency and Load-Balancing

As mentioned in the introduction, sensor nodes are energy constrained. In this section we present methods for implementing load-balancing mechanisms using the multiple paths, created by the proposed algorithm in the prior sections.

As we know, most of energy utilized in a sensor node is usually consumed for the routing procedure especially in large scaled sensor networks where the distance between the source of data and the destination is significant. So it is very important to spread this overhead between different nodes. In this paper the multi-path routing method, used mostly for load-balancing. Although constructing a couple of disjoint or partially-disjoint paths between sink and source will lead to longer life-time of each path and also the connection will be more robust against failures. Therefore, the rate of interest flooding can be reduced which is an expensive operation due to its large communication overhead although this subject is beyond our work. In this paper we will focus on load-balancing between multiple paths created by the proposed multipath routing algorithms.

A simple approach to perform load-balancing is spreading data packets uniformly between different paths. Then PATH_ALIVE messages can be sent periodically through each path. If a path becomes unusable due to energy consumption of its nodes or other reasons, this path will not be selected anymore.

In this paper we present a more efficient approach in which the *path minimum energy (e)* and the *path length (l)* are regarded during the packet spreading process between the alternate paths.

Path minimum energy, e is defined as the energy of node with the least amount of energy between the nodes along the path and the *path length* is defined as the number of hops in that path. For reinforcing the paths with bigger amounts of *e*, we consider the probability of sending a packet along a path, directly proportional to its minimum energy. Let $e_p(i)$ be the path minimum energy of path with MP_ID *i* and $P_{sel}(i)$ be the probably of selecting path *i*. In this case:

$$P_{sel}(i) \propto e_p(i) \tag{1}$$

Another parameter used is the path length *l*, as the path length grows, the cost of routing along that path would be increased linearly by the length of it. So we prefer to select the paths with shorter lengths. So:

$$P_{sel}(i) \propto \frac{1}{l_p(i)} \tag{2}$$

By this approach, at first, shorter paths have more chance to forward the data packets but after a period of time, their energy would be decreased. In this situation the longer paths with more energy, would have a better chance of being selected by the source node.

In order to further improve this we introduce two thresholds named as *minimum energy threshold* (e_{th}) and *maximum path length* (l_{th}). When the energy of a path reaches blow e_{th}, or its length is more than l_{th}, the probability of its selection by the source node, calculated according to the formulas (1) and (2) multiplied by 0.1 and 10. In our simulations, we use the l_{th} with value $l_{min}*\beta$ and the eth is selected as e_{max}/α where l_{min} is the length of the path with the minimum length and e_{max} stands for the *e* (minimum energy) of the path with the maximum *e*. So the probability for the source node to select a path between multiple paths is calculated by:

$$P_{sel}(i) = c\,\frac{e(i)}{l(i)} \tag{3}$$

And *n* is the number of paths constructed by the routing algorithm and *α* and *β* are threshold factors. Here we assumed these values to be *α* = 2 and *β* = 4 according to our simulation experiments.

$$\begin{gathered} e(i) = \begin{cases} e_p(i) & e_p(i) > e_{th} \ and\, n > o \\ 0.1 * e_p(i) & e_p(i) < e_{th} \end{cases} \quad , e_{th} = \max[e_p(i)] \,/\, \alpha \\ l(i) = \begin{cases} l_p(i) & l_p(i) < l_{th} \ and\, n > o \\ 10 * l_p(i) & l_p(i) > l_{th} \end{cases} \quad , l_{th} = \min[l_p(i)] \times \beta \end{gathered} \tag{4}$$

$$c = \frac{1}{\sum_{i=1}^{i=n} \frac{e(i)}{l(i)}} \tag{5}$$

The e_{th} changes dynamically during the routing period. To inform the source node, about the energy of each path, periodic PATH_ALIVE messages are sent along every path. The PATH_ALIVE messages that used for path refreshment are the same as POSITIVE_REINFORCEMENT messages and calculate the minimum energy of the paths. Using this method, the connection time between source and sink can be increased and the number of packets dropped during the routing process can be reduced. Another way to increase the energy efficiency of routing protocol is using the LFI method that reduces the overhead of flooding ED packets. As we will see in the section 6, this can increase lifetime of the network significantly.

5 Methodology

We simulated DD algorithm available with ns-2 simulator version 2.30. This protocol is implemented for simulator in two versions. We used diffusion 3 protocol [3] which is a complete protocol implementation and allows a more realistic evaluation of the protocol.

The traditional DD algorithm floods an interest message every 30 seconds and exploratory data every 50 seconds. We implemented two kinds of simulations. The first one uses the standard parameters used in the original DD algorithm. In this simulation, we used the ping application as the network traffic with the rate of 10 packets per second. In the other simulation, we used the ping traffic with the interval of 5 seconds (as the original application) but we assumed the interest flooding interval and the ED packets flooding interval to be 300 seconds and 500 seconds respectively in order to highlight the routing overhead, imposed by different multi-path construction algorithms.

In the original directed diffusion, the IEEE 802.11 is used for the MAC layer. For comparability we used the same MAC layer and energy model as in [6] that is the PCM-CIA WLAN card model in ns-2. This card consumes 0.660 W when sending and 0.395 W when receiving. In this paper the transmission range is assumed to be fixed and 200 meters.

The protocols are tested in a 10*10 grid (100 nodes), with one sink and one source. Each protocol has been simulated at least five times and the mean value of each measurement has been considered. This iteration was quite necessary, for the random behavior of most proposed algorithms. Then we measured the following parameters: number of paths, total number of share nodes among the paths, average minimum and maximum path length and finally mean path length. We used four densities to highlight the effect of density of nodes on the results of different path construction algorithms.

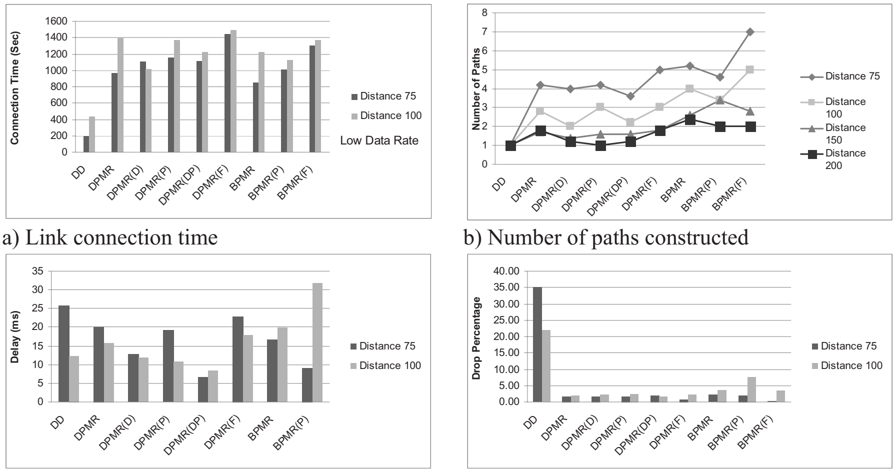
For the measurement of energy-efficiency and studying the effect of load-balancing effects on increasing the life time of permanent connections between nodes, we used the two scenarios, presented in section 4.B. We assumed the initial energy of all nodes in the network to be 5 joules.

In our scenario, the source starts to send ping data packets towards the sink continually, until the connection is broken, due to path node failures caused by energy depletion. This period is measured and considered as connection life-time. The average delay is also calculated for each routing algorithm.

In order to calculate protocol overhead, we measured the number of none-data packets, after receiving 100 data packets (in the sink node). Our measurement may not be quite precise but it helps us to have a slight comparison between the overhead, imposed by different algorithms.

6 Simulation Results

In this section we will show the simulation results, achieved by implementing the scenarios and assumptions, described in the last section.



a) Link connection time

b) Number of paths constructed

c) Delay

d) Drop Percentage

Fig. 3. (a) Link connection time (b) Number of paths constructed (c) Delay (d) Drop Percentage

A. Multipath Construction

In figure 3.a, the number of paths, constructed by different approaches has been shown for different algorithms. Two main results can be extracted from this chart. First the number of paths produced by the different multi-path algorithms grows by increasing the density of the network. Second, among the presented methods, the BPMDD-F, PMDD-F and BPMDD-P algorithms can produce more paths than the other methods. As PMDD-F method constructs disjoint paths, it is more robust to failures.

B. Connection Time and Overhead Estimation

Using multi-path routing and load-balancing approaches will increase the connection time between source and sink. As shown in figure 3.b, multi-path routing has a prominent effect of the connection life-time. The PMD-F performs better than the other algorithms proposed. This is due to the limited forwarding improvement of this approach.

The overhead of the directed diffusion will decrease by using RFI and LFI methods. Especially the PMD-F algorithm shows a significant reduction in routing protocol overhead (Figure 4).

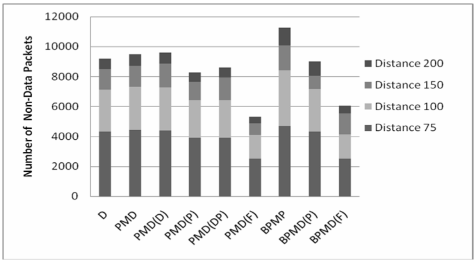


Fig. 4. Overhead Comparison between Different Multipath Methods

C. Delay Computation and Drop Percentage

As it is shown in figure 3.c, the multi-path routing methods, will reduce the delay in some methods as PMD-DP but generally most of these methods increases the delay. The drop percentage is improved significantly using multi-path methods as depicted in figure 3.d.

8 Conclusion

This paper describes the use of multipath routing for implementing load-balancing and increasing the energy efficiency of the routing algorithm. In this work four methods were proposed for constructing multiple paths between sinks and sources: SMDD, DPMDD, BPMDD and SPMDD and also four different approaches were presented to improve the efficiency of such algorithms: RFI, RDI, LFI and HI.

Simulation results show that the BPMDD and DPMDD algorithms when used with the LFI approach, have better performance than the other methods. The LFI method also decreases the number of exploratory data packets flooded in the network, significantly and reduces the overhead of two-phase pull algorithm but this method cannot be adapted to one-phase-pull algorithms due to its limited forwarding approach. And for multipath routing in one-phase-pull algorithm, DPMDD method can be adapted with RFI improvement approach. This can simply performed just by changing the roles of exploratory data packets in two-phase-pull algorithm by interest packets in one-phase-pull. Also the number of paths produced by our routing algorithm grows by increasing the density of the nodes in the network. The simulation results also show that using multipath routing algorithm can lead to longer connection-life and more energy efficient routing.

For future work, we consider two directions. First we try two implement the proposed multipath approaches for the one-phase-pull algorithm and find an efficient algorithm. Second, we will improve our multi-path selection method to prefer node with better energy resources to others and produce more energy efficient paths.

References

[1] Intanagonwiwat, C., Govindan, R., Estrin, D., Heidemann, J., Silva, F.: Directed diffusion for wireless sensor networking. ACM/IEEE Transactions on Networking 11(1), 2–16 (2002)

[2] Heidemann, J., Silva, F., Intanagonwiwat, C., Govindan, R., Estrin, D., Ganesan, D.: Building Efficient Wireless Sensor Networks with Low-Level Naming. In: Proceedings of the Symposium on Operating Systems Principles, pp. 146–159 (October 2001)

[3] Heidemann, J., Silva, F., Yu, Y., Estrin, D., Haldar, P.: Diffusion filters as a flexible architecture for event notification in wireless sensor networks. Technical Report ISI-TR-556, USC/Information Sciences Institute (April 2002)

[4] Heidemann, J., Silva, F., Estrin, D.: Matching Data Dissemination Algorithms to Application Requirements. In: Sensys 2003. The First ACM Conference on Embedded Networked Sensor Systems, pp. 218–229 (November 2003)

[5] Ganesan, D., Govindan, R., Shenker, S., Estrin, D.: Highly Resilient Energy-efficient Multipath Routing in Wireless Sensor Networks. In: Proceedings ACM MOBIHOC, pp. 251–253 (2001)

[6] Handziski, V., Köpke, A., Karl, H., Frank, C., Drytkiewicz, W.: Improving the Energy Efficiency of Directed Diffusion Using Passive Clustering. In: Wireless Sensor Networks, First European Workshop, pp. 172–187 (January 2004)

Flattening the Gap Between Source-Destination Paths in Energy Efficient Greedy Georouting in Wireless Sensor Networks

Essia H. Elhafsi and David Simplot-Ryl

Univ. Lille 1, CNRS/INRIA, France
essia@cs.ucr.edu,simplot-Ryl@lifl.fr

Abstract. In this paper, we are interested in simple routing protocols, for wireless sensor networks. One of the major challenging issue in sensor networks is that nodes rely on batteries with limited capacity. Therefore, to extend the life of the sensors and the reliability of the network, an efficient routing algorithm can reduce/optimize energy consumption by reducing routing overhead while maintaining a high delivery rate. Moreover, a good routing method should compute a routing path that is as close as possible to the shortest path. We propose *ORouting*, a localized energy efficient greedy routing scheme. To minimize the gap between the shortest path and the computed path, ORouting computes its routing path by locally selecting the next hop node based on its orthogonal distance to the direction of the source/destination pair. Moreover, in its routing decision, ORouting is biased towards its neighbors in the forward direction towards the destination. We compare ORouting to several routing protocols. We show, through simulation, that our protocol improves significantly energy consumption and achieves a high percentage of successful routings.

1 Introduction

Wireless sensor multiphop networks are defined by nodes communicating without using a fixed infrastructure. Each mobile host can communicate with other hosts within its transmission range r. However, due to limited communication ranges, sending a message from a source to a destination which are not within range of each other, often requires the collaboration of intermediate forwarding nodes.

In sensor networks, nodes rely on batteries with limited capacity. Therefore, the most important criteria when designing communication protocols is to reduce the communication overhead to optimize their energy consumption and extend the life of the sensor device. Consequently, we extend the connectivity and reliability of the underlying network. In this framework, localized routing protocols based on geographic information of the sensors have been proposed as a viable alternative to existing routing protocols (such as AODV [13] and DSR [9]) for wireless networks in order to reduce the overhead of maintaining routing tables in the sensors and to avoid the cost (energy consumption) of flooding and route discovery.

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 56–65, 2007.

The literature in regards to general routing in sensor network is rich [16,17,11, 15,10]. Geographic routing (georouting) forms a specific class of routing protocols which requires each network node to be able to determine its coordinates by means of a location system like GPS [6], by relative positioning based on signal strength estimation [7] or other means [5].

The notion of progress is a key concept of several georouting protocols proposed for sensor networks. Given a transmitting node S and a destination node D, the progress of a node A is defined as the projection onto the line connecting S and D. The distance between S and the receiving neighbor node A is in forward direction if the progress is positive (for example, in Figure 1 nodes A, B and C are in forward direction while nodes E and F are in backward direction). Basic distance, direction and progress methods use these concepts to select the neighbor to forward a message. Greedy routing protocols fall within this class of routing schemes.

In this work, we are interested in energy-aware georouting algorithms. Ideally, to achieve this objective, a message should be sent using the "best" (shortest $\equiv$ (SD) line in Figure 1) path. However, since we are interested in localized algorithm, we compute paths as close as possible to the shortest path. We propose a simple localized georouting scheme that minimizes energy by reducing the gap between the shortest path and the computed path. If we consider the triangle constructed from the source S, the destination D and a chosen intermediate node X, we claim that by reducing the area of this triangle, we compute a path very close to the shortest path. Given that the distance from the source to the destination is constant, reducing the area of triangle (S, D, X) is equivalent to reducing its height h which we define as the gap between the computed path and the shortest path. Therefore, to minimize energy we have to "flatten" or minimize this gap.

This paper is organized as follows. In Section 3, we present a brief summary of greedy routing protocols relevant to this work; Most forward routing (MFR) which key progress is based on a distance measure and Compass routing which decision progress is based on direction. We also present a Cost over progress routing protocol, which is known to be an energy-efficient scheme. In Section 4, we present an energy-aware greedy routing scheme, Orthogonal Routing or *ORouting* which key progress is a combination of distance measure and direction. In Section 5, we compare the performance of our proposed method to alternative routing algorithms presented in Section 3. Finally we conclude in Section 6.

2 Network Model

While a network model can be arbitrary, the simulations are generally based on the widely adopted *unit disk graph [8]* (UDG). A UDG is defined over a graph $G = (V, E)$ where V is the set of sensor nodes and $E \subseteq V^2$ is the set of edges which allow the available communications. Let A and B be two nodes of the graph with a transmission range $\mathrm{r_A}$ and $\mathrm{r_B}$ respectively. If we let $|AB|$ be the Euclidean distance between nodes A and B, then the set E is defined as follows:

$E = \{(A, B) \in V^2/|AB| \leq r\}$. Nodes A and B are said to be neighbors (thus joined by an edge) in the network and can communicate if the Euclidean distance is less than the minimum of their transmission ranges ($|AB| \leq min(r_A, r_B)$). Finally, We define the neighborhood set $N(A)$ of a node A as: $N(A) = \{B \in V|B \neq A \wedge (A, B) \in E\}$.

In this work, we assume that nodes in the network are equipped with GPS receivers and are aware of their geographic coordinates and the ones of their one hop neighbors.

3 Greedy Routing Protocols

Greedy routing schemes are the simplest algorithms to use in wireless networks. Several protocols have been proposed [10,3,4] . In this work, we classify greedy routing based on their routing progress metrics namely hop count and energy consumption based routing.

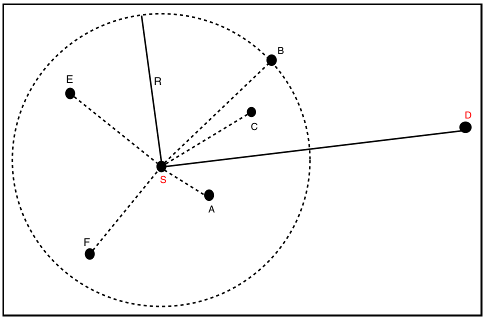


Fig. 1. Greedy schemes when node S with transmission range r has a packet to send to node D: MFR (to B) and Compass routing (to C)

3.1 Hop Count Based Progress

In [16], a simple greedy routing, that we refer to as *Most Forward Routing (MFR)* is proposed. Each node is assumed to know the position of all its neighbors within a distance r of its position. Given a packet and its destination, a node transmits to the node most forward (among those whose position it knows) in the direction of the final destination. If no node is in the forward direction, it transmits to the least backward node, if any. In case there are no nodes in the circle of radius r, the packet is dropped. This algorithm works well in a dense graph. In [14], Stojmenovic and Lin show that MFR algorithm is probed to be loop-free. Moreover, it is the only progress-based algorithm competitive in terms of hop-count.

In [11] Compass routing is proposed. The source or intermediate node S uses the location information for the destination node D to calculate its direction. The location of the one hop neighbors of S is used to determine to which of them, say node C, the direction SC is closest to the direction SD (that is $\widehat{CSD}$ is minimized). Once a node is selected this process repeats until the destination node is reached (or the routing fails). In Figure 1, the direction SC is closest to the direction SD among candidate directions SA and SB. The path selected is then SCD. In general, this method is not loop-free. This property holds only in the case of planar graphs with convex regions [14]. Moreover, due to its indeterminism, Compass routing generates quite long paths in practice.

3.2 Energy Based Routing

In [12] a localized energy aware algorithm is proposed where nodes are equipped with GPS receivers. Each node makes a routing decision on the basis of its location, its neighbors and the destination. A node forwards the packet to the neighbor closer to the destination such that the ratio of the energy consumed to the progress made (measured as the reduction in distance to destination) is minimized. To route a packet to destination D, a node extracts the coordinates of D from the packet and chooses a forwarding node in its neighborhood. The idea is that the current node X chooses node $Y \in N(X)$ which minimizes $\frac{\mathrm{cost}(X,Y)}{\mathrm{progress}(X,Y,D)}$ where $\mathrm{cost}(X,Y)$ represents the power consumed to send the message form X to its neighbor Y, and where $\mathrm{progress}(X,Y,D)$ is the progress in the routing task which can be expressed as the difference $|XD| - |YD|$.

The most common energy model is proposed in [1], $power(r) = r^{\alpha} + c$ f $r \neq 0$, 0 otherwise. where r is the distance separating two neighboring nodes a forwarding and a receiving node; c is the overhead due to signal processing; α is a real constant greater than 1 and it represents the signal attenuation and depends on the wireless environment. In this work, we ignore the energy consumed due to signal emission. In [15] the optimal transmission radius, r^*, that minimizes the total power consumption for a routing task is computed and is equal to: $\mathrm{r}^* = \sqrt[\alpha]{\frac{\mathrm{c}}{\alpha-1}}$ if nodes can ideally be placed on a line toward the destination.

For this protocol to work, the current node has to limit its choices to neighbors with positive progress. To the best of our knowledge, cost over progress is the best performing protocol in terms of energy savings.

3.3 Pit-Falls of Greedy Routing

There are several scenarios in which greedy routing protocols fail. For instance, if the mobile host w_i is closer to the destination than any of its neighbors, or if two adjacent mobiles w_{i-1} and w_i are equally close to the destination, and none of their other neighbors is closer, then a deadlock results and the packet never reaches its final destination. Moreover, greedy routing are prone to be loop-free, thus increasing energy consumption without reaching the ultimate goal and delivering a message to the final destination. Finally, greedy routing like MFR and Compass are not energy aware. Their main objective is to reach the destination.

4 Contribution

We propose a greedy routing protocol that we refer to as *ORouting*. We show that ORouting is more efficient than Compass routing and MFR in terms of energy savings and delivery rate.

Our algorithm combines and exploits both MFR and Compass routing objectives. MFR chooses the next hop neighbor based on the latter's distance to the destination while Compass routing chooses the next hop neighbor, that has the closest direction to the line connecting the source to the destination.

ORouting assumes that each node is aware of its geographic location and the location of its one hop neighbors. It chooses the next hop neighbor (to relay the packet) that has the smallest orthogonal distance to the line connecting the transmitting node (which may be the source node or any other node) and the destination node. This process repeats until the destination is reached or the routing fails.

Our algorithm is energy aware. Let X be a transmitting node and let n be the number of its one hop neighbors. To each one hop neighbor $Y_i \in N(X), 0 < i \leq n$, ORouting associates a penalty factor, computed as a function of its distance to X and the optimal transmission range r* (given by Equation **??**) as follows. If the distance between Y_i and X, $|XY_i|$, is within range of r*, the choice of Y_i incurs no penalty, otherwise it incurs a penalty c_i that is equal to $|XY_i|/\mathrm{r}^*$ (which is obviously greater than 1). This penalty factor insures that the selected nodes have a distance to the transmitting node as close as possible to the optimal transmission range r*.

If we let h_i be the orthogonal distance of neighbor Y_i to line (XD), the objective of ORouting can then be formulated as follows,

$$min \quad c_i \cdot h_i, \quad 0 < i \leq n$$

where,

$$c_i = \begin{cases} 1 & \text{if } |XY_i| \leq \mathrm{r}^*, \\ \\ |XY_i|/\mathrm{r}^* & \text{otherwise} \end{cases} \qquad (1)$$

In the case when X has no neighbors ($i = 0$), the message is dropped.

The motivation behind this scheme is the following. We claim that the path computed (and consequently the cost or energy consumed) to reach the destination is closely related to the area of the triangle with vertices the transmitting node X, the destination D and the next hop neighbor Y_i. If we let h_i be the height of such a triangle, then the area of the triangle is equal to $\frac{h_i \cdot |XD|}{2}$. Since for a given node X, $|XD|$ is a constant independent of the selected neighbor Y_i, then minimizing the area of the triangle reduces to minimizing h_i which is nothing but the orthogonal distance of the next hop neighbor Y_i to the line (XD). Clearly, the optimal (shortest) route to reach the destination is to follow a path on line (SD) (if it exists). By minimizing h_i, ORouting tries to find the path that is as close as possible to the shortest path. If $h_i = 0$, ORouting computes the shortest path assuming that there are nodes along line (SD).

As an example, consider node S in Figure 2. If we denote by $h_a = dist(A, (SD))$, by $h_b = dist(B, (SD))$ and by $h_c = dist(C, (SD))$ (where $dist(T, (SD))$ is the orthogonal distance from T to line (SD) for an arbitrary node T) then since $h_a \leq h_b$ and $h_a \leq h_c$, then S selects node A to relay the packet. For simplicity, we assume that nodes A, B and C are within range of r* ($\mathrm{r}^* \geq max(|SA|, |SB|, |SC|)$). Node A will repeat the same process where the segment SD is now replaced by segment AD. Clearly, for this simple example the route computed by ORouting is SAD. The routes computed by MFR and Compass routing on the other hand are respectively SBD, and SCD.

ORouting drops the message if the best choice for the current node is to return the message to a previously visited node (a node a message came from). Moreover, since the neighbors considered in the choice are the ones in the forward direction, it can be easily verified that this algorithm is loop-free (*i. e.,* a source of energy waste is eliminated).

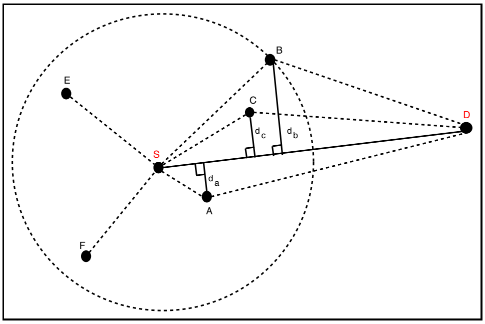


Fig. 2. Orthogonal greedy routing scheme when node S with transmission range r has a packet to send to node D: node A is relayed the packet since it is the closest node to the line connecting the source destination pair

5 Experimental Results

We compare the routing protocols based on the following performance measures: delivery rate and energy consumption. The delivery rate is measured as the proportion of time the packet reaches its final destination. To simplify our system, we use our own C simulator that assumes an ideal MAC layer, *i.e.* no interferences and no packet collisions. In the simulated network we assume that nodes are Poisson distributed over a 2-dimensional space with the average number of nodes per unit area equal to λ. Moreover, we assume that these nodes have the same transmission range, r = 0.1.

Our simulation results show that all routing schemes guarantee delivery for dense networks ($500 < \lambda \leq 1000$). For less dense networks, Figure 3 shows that

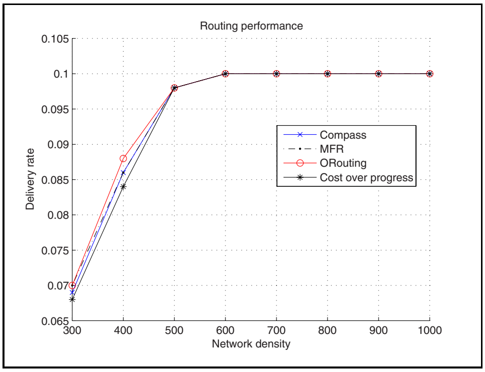


Fig. 3. Delivery rate achieved by MFR, ORouting, Compass and Cost over progress protocols

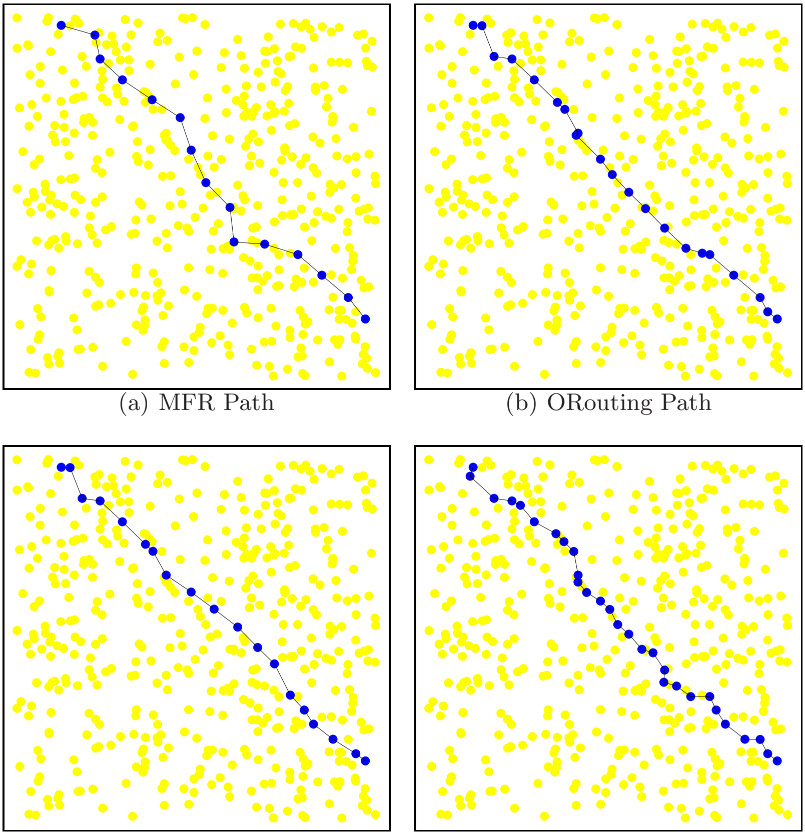
ORouting outperforms MFR, Compass and Cost over progress protocols, Cost over progress being the least performing and MFR being the next best protocol. Figure 4 shows sample paths computed by the various protocols for $\lambda = 500$.

To test the routing protocols in regards to their energy efficiency we rely on the energy model described in subsection 3.2 where we assume that $\alpha = 4$ and $c = 10^8$, and therefore $\mathrm{r}^* = 0.04$.

We compare the protocol's energy efficiency based on the overall energy consumed due to routing successes and to routing failures. Figure 5(a) shows that ORouting outperforms MFR and Compass routing; MRF being the worst performing scheme. Figure 5(b) describes in more details the average energy consumption per hop. The main reason of this result is explained by Figure 6(a) which shows that the edges computed by ORouting are shorter than the ones computed by MFR and Compass routing and are closer to the optimal transmission range r^*. Consequently, we can infer that ORouting computes paths with larger number of hops than MFR and Compass routing (see Figure 6(b)). MFR, however, follows the longest edges thus computes paths with low number of hops. Cost over progress, on the other hand, outperforms all protocols.

Mathematically, we can prove that ORouting guarantees that the path length (in Euclidean distance) is shorter than the paths computed by the other greedy routing algorithms. Consider Figure 2, the route computed by MFR is SBD, its length is $L_{MFR} = |SB| + |BD|$, the one computed by Compass routing is SCD, its length is $L_{Compass} = |SC| + |CD|$, and finally for ORouting the route is SAD, its length is $L_{ORouting} = |SA| + |AD|$. Clearly, $L_{MFR} > L_{Compass} > L_{ORouting}$[1]. Figure 4 confirms the above claim. It shows that the path followed by ORouting is very close to being a linear path compared to the Alternatives.

[1] The proof follows by computing the area of the following triangle (S, B, D), (S, C, D) and (S, A, D).



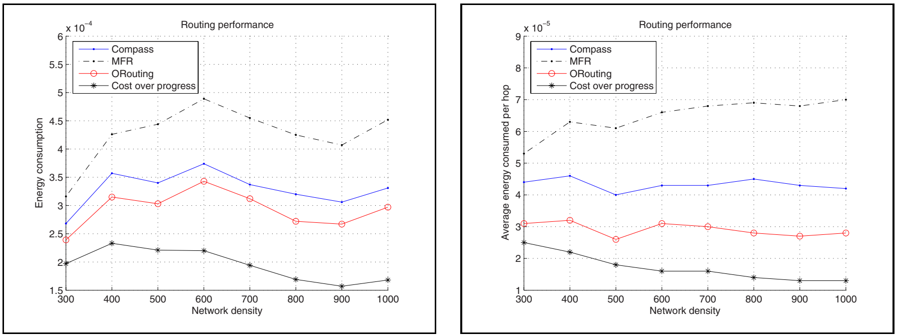
(a) MFR Path

(b) ORouting Path

(c) Compass Path

(d) Cost over Progress Path

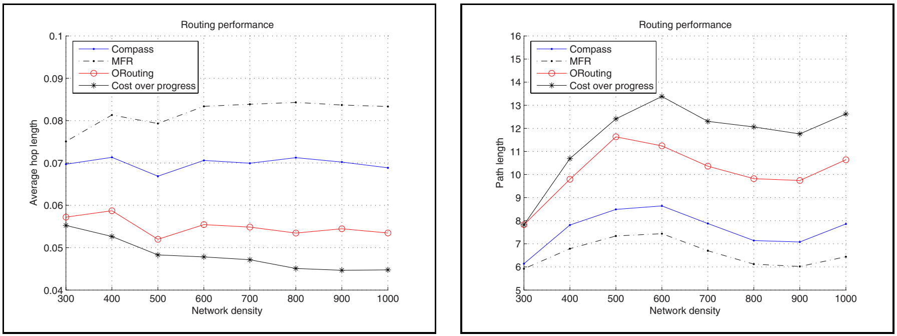
Fig. 4. Path followed between a pair of nodes by the routing protocols $\lambda = 500$



(a) Overall energy consumption

(b) Energy consumption per hop

Fig. 5. Performance of the routing protocols–energy efficiency



(a) Average hop length

(b) Average path length

Fig. 6. Performance of the routing protocols–characteristics of the computed paths

6 Conclusion

In this work, we propose ORouting a localized, greedy routing protocol. We show through simulation that our protocol improves significantly energy consumption and achieves a high percentage of successful routings in arbitrary network topology. We compare ORouting to MFR, Compass and Cost over progress routing protocols. Our method outperforms the greedy hop count based routing protocols in terms of energy efficiency and delivery rate. However, it outperforms Cost over progress only in delivery rate. We also show that ORouting includes Compass and MFR as special cases.

References

1. Rodoplu, V., Meng, T.: Minimizing energy mobile wireless networks. IEEE Journal on Selected Areas
2. Karp, B., Kung, H.T.: GPSR: Greedy perimeter stateless routing for wireless networks. In: Proc. MOBICOM, pp. 243–254 (2000)
3. Li, J., Gewali, L., Selvaraj, H., Muthukumar, V.: Hybrid greedy/face routing for ad-hoc sensor network. dsd 0, 574–578 (2004)
4. Frey, H., Stojmenovic, I.: On delivery guarantees of face and combined greedy-face routing algorithms in ad hoc and sensor networks. In: Twelfth ACM Annual Int. Conference on Mobile Computing and Networking MOBICOM, Los Angeles, pp. 390–401 (September 2006)
5. Elhafsi, E.H., Mitton, N., Simplot-Ryl, D.: Cost over progress based energy efficient routing over virtual coordinates in wireless sensor networks. In: t2pWSN 2007. Proc. IEEE International Workshop: From Theory to Practice in Wireless Sensor Networks, Helsinki, Finland (2007)
6. Niculescu, D., Nath, B.: Ad hoc positioning system (APS). In: Proceedings of GLOBECOM, San Antonio (2001)
7. Hightower, J., Borriella, G.: Location systems for ubiquitous computing. IEEE computer 8(34), 57–66 (2001)

8. Clark, B.N., Colbourn, C.J., Johnson, D.S.: Unit disk graphs. Discrete Math. 86(1-3), 165–177 (1990)
9. Johnson, D., Maltz, D., Broch, J.: DSR The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks. ch.5, pp. 139–172. Addison-Wesley, Reading (2001)
10. Karp, B., Kung, H.T.: Gpsr: Greedy perimeter stateless routing for wireless networks. In: Proc. MOBICOM, pp. 243–254 (2000)
11. Kranakis, E., Singh, H., Urrutia, J.: Compass routing on geometric networks. In: Proc. 11 th Canadian Conference on Computational Geometry, Vancouver, pp. 51–54 (August 1999)
12. Kuruvila, J., Nayak, A., Stojmenovic, I.: Progress and location based localized power aware routing for ad hoc sensor wireless networks. IJDSN
13. Perkins, C.: Ad-hoc on-demand distance vector routing. In: MILCOM (November 1997)
14. Stojmenovic, I., Lin, X.: Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks. IEEE Trans. Parallel Distrib. Syst. 12(10), 1023–1032 (2001)
15. Stojmenovic, I., Lin, X.: Power-aware localized routing in wireless networks. IEEE Trans. Parallel Distrib. Syst. 12(11), 1122–1133 (2001)
16. Finn, G.: Routing and addressing problems in large metropolitan-scale. Internetworks, ISI Research Report ISU/RR-87-180 (March 1987)
17. Takagi, H., Kleinrock, L.: Optimal tarnsmission ranges for randomly distributed packet radio terminals. IEEE transaction on communications com-22(3), 246–257 (1984)

Adaptive Topology Based Gossiping in VANETs Using Position Information[⋆]

Boto Bako, Igor Rikanovic, Frank Kargl, and Elmar Schoch

Ulm University, Institute of Media Informatics
{givenname.surname}@uni-ulm.de

Abstract. Gossiping is a lightweight and simple technique for information dissemination in many application domains, be it in Wireless Sensor Networks (WSNs), Mobile Ad-hoc Networks (MANETs), or Vehicular Ad-hoc Networks (VANETs). Much research has been conducted in the past on probabilistic dissemination methods because of their efficiency compared with simple flooding and their simple application. However most work was focused on static gossiping, i.e., the gossiping probability cannot be adapted according to topology changes. Thus, topology characteristics have to be known in advance.

In this work the use of position information for building up a neighborhood relationship is proposed. Based on this information, a forwarding hierarchy is constructed and the protocol is capable to adjust the dissemination probability dynamically in a distributed manner. The protocol is evaluated in a highway scenario, where the network characteristic varies from sparse networks with highly mobile nodes to a traffic jam with very high node density and low node velocities. The applicability of the proposed protocol for such scenarios is shown by simulations.

Keywords: Information dissemination, gossiping, probabilistic broadcasting, vehicular ad hoc network (VANET).

1 Introduction

Controlled dissemination of information plays an important role in many network scenarios like wireless sensor or vehicular ad-hoc networks. A message is broadcasted to all nodes in a network to fulfill application specific services or to enable some basic mechanisms for routing. In sensor networks it is used e.g. to disseminate TAG-type queries [1], to broadcast control messages and so on. Flooding is the simplest form of broadcasting and is used also in many routing protocols, e.g. in DSR [2] and in AODV [3] to establish routes. Also protocols in the automotive domain use flooding, as CGGC [4] for example for area forwarding. However, flooding works well for a small number of nodes but leads to high performance problems in dense networks regarding channel congestion and packet collisions. This leads the so called "broadcast storm problem" [5]. As

[⋆] This work has been supported by a grant from the Ministry of Science, Research and the Arts of Baden-Wrttemberg (Az: 23-7532.24-14-19) to Boto Bako.

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 66–78, 2007.

simple flooding wastes a lot of bandwidth, more efficient message dissemination schemes have been developed.

Gossiping is one widely used dissemination technique because of its robustness and scalability. The simplest variants use a global probabilistic value at all nodes and are thus called static gossiping (confer, [6], [7], [8]). All these approaches work only if the network topology is known in advance otherwise it will result in a low delivery ratio or a high number of redundant messages. To overcome these problems adaptive gossiping has been introduced.

Haas et al. [7] introduced the so called two-threshold scheme, an improvement for static gossiping based on neighbor count. A node forwards a message with probability $p1$ if it has more than n neighbors. If the number of neighbors of a node drops below this threshold n then messages are forwarded with a higher probability $p2$. The obvious advantage of this improvement is that in regions of the network with sparse connectivity messages are prevented to die out because the forwarding probability is higher than in dense regions.

[7] also describes a second improvement which tries to determine if a message is "dying out". Assuming a node has n neighbors and the gossiping probability is p then this node should receive every message about $p \cdot n$ times from its neighbors. If this node receives a message significantly fewer, the node will forward the message unless it has not already done so.

In [5], Ni et al. introduced the Counter-Based Scheme. Whenever a node receives a new message, it sets a randomly chosen timeout. During the timeout period a counter is incremented for every duplicate message received. After the timeout has expired, the message is only forwarded if the counter is still below a certain threshold value.

Although all these adaptations improve the broadcast performance, they still face problems in random network topologies. For example, a node may have a huge number of neighbors, resulting in a small forwarding probability in all of these schemes. Despite this, there could e.g. still be a neighbor which can receive the message exclusively from this node. An example of such a situation is shown in Figure 1 (example taken from [9]).

When node A sends a message, all nodes in its neighborhood receive it. In this example scenario only node E should forward it with the probability of 1 since E is the only node that can propagate the message to node G. If the gossiping probability is only based on the neighbors count, node E will be assigned a low probability since it has many neighbors. So the broadcasted message will "die out" with a high probability and never reach G and all later nodes. If the part

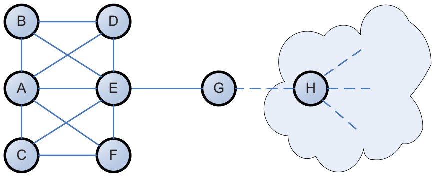


Fig. 1. Sample topology where static gossiping fails

of the network connected only via G is very large, the overall delivery ratio will drop dramatically. Such situations can occur quite regularly in dynamic networks of a certain density.

Recently, the Smart Gossip protocol which addresses this problem was introduced by Kyasanur et al. [9]. The protocol assumes a static network with one single message source. Every node in the network uses neighborhood information from overheard messages to build a dependency graph. I.e., each node has a parent, sibling, and child set. Parents are nodes where this node receives new messages from, siblings receive messages from the same parents and the node delivers messages to child nodes.

Depending of the number of parents, every node calculates the probability by which its parents should forwards a message and informs its parents about this probability. A parent sets its forwarding probability to the maximum of all child probabilities. If a node has only one parent, the forwarding probability will be automatically set to 1, if a node has many parents, the probability will be comparatively small, but still large enough to ensure that the node will most likely receive the message at least once.

Smart Gossip solves the problem of adapting the forwarding probability dynamically, but in some cases there are still disadvantages: the described parent child relationship is dynamic, depending on which node sends or forwards a message. For ensuring to build up a stable directed graph, the authors make the assumption that there is only one message originator in the whole network. This assumption however cannot be fulfilled in our scenario. We want to apply a gossiping like message dissemination for vehicular ad hoc networks (VANETs) where the nodes are moving vehicles. VANETs have quite different properties compared e.g. with sensor networks (WSNs). In the upcoming section we give a brief overview over these characteristics and discuss their impacts to gossiping.

2 Scenario Description and Assumptions

VANETs represent highly dynamic networks. Due to the high mobility and speed of vehicles, the network topology changes rapidly. Two vehicles might be in mutual communication range only for a few seconds. Additionally, the properties like speed or movement patterns can vary significantly within VANETs.

On a free highway for example, vehicles can move with high velocity, whereas, on the same highway in a traffic jam the number of vehicles is extremely high and the velocity drops to nearly zero. In the first case there is a highly mobile and sparse network. Nodes are moving with high speeds, neighborhood changes constantly and the network is often partitioned. For this case because of the low density the message dissemination algorithms needs to fall back to flooding and forward the messages in every node.

On the other hand, the challenge in very dense networks like in traffic jams is to avoid channel congestion through an effective broadcasting algorithm. In such dense networks the overall node mobility is low, thus neighborhood changes are infrequent and the topology is almost static.

As an application example, consider a traffic information system where vehicles detect traffic density and report this information to other vehicles so they get a complete picture of the traffic jam. This information needs to be disseminated in a certain area and the message has to reach all vehicles within this area if possible. At the same time the number of retransmissions has to be minimized to avoid a broadcast storm.

In contrast to WSNs, we assume that energy consumption is not an issue. The vehicles will provide the necessary power and computing capacity. Another likely assumption is that all VANET-enabled vehicles will be equipped with a GPS receiver, thus they can determine their positions. This information is used in the proposed protocol to achieve effective message dissemination. There is another important difference from WSNs: there typically one dedicated node broadcasts e.g. control messages into the network. In the mentioned traffic jam scenario, it is quite obvious that every vehicle could be the originator of messages which inform other vehicles about its own status, detected road condition, traffic density, etc..

The Smart Gossip protocol is a promising approach for efficient message dissemination in WSNs but will fail in the presented scenario. These drawbacks are discussed in the next section.

3 Review of Smart Gossip with Multiple Originators

As mentioned before, one of the assumptions of the Smart Gossip protocol is that only one node in the network initiates broadcasting messages. This assumption is required to build up a correct parent-sibling-child relationship. The hierarchy is established by overhearing of normal gossip messages. The initial gossip probability is 1 and is adjusted step by step as more topology information becomes available.

Based on the sample topology in Figure 2 we will show where the original Smart Gossip algorithm fails, if there is more than one originator.

We assume that nodes A and D are originators of gossip messages. If node A and D begin to send messages, the nodes B and C receive these messages and include A and D in their parent sets. According to the Smart Gossip terminology B and C are children of nodes A and D, A and D are parents of B and C, and B and C are siblings.

Because B and C have no children themselves, they need not forward a received gossip message. In the Smart Gossip protocol they nevertheless forward

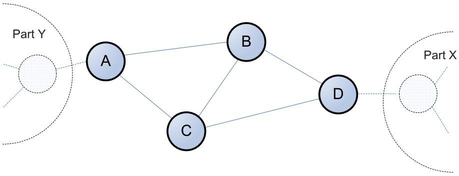


Fig. 2. Sample topology where Smart Gossip fails with multiple originators

such messages with a low probability which achives a higher robustness. In this situation, messages which are created in network part X will likely be dropped by B and C and will most likely never reach part Y and vice versa. Hence, the average reception percentage in such topologies drops significantly. Although Figure 2 shows an artificially created topology, such node distributions can occur frequently in sparse areas of dynamic networks.

As we have shown, for vehicular networks we need a dissemination protocol which can cope with multiple message originators and therefore considers the direction of gossip messages while creating the hierarchy. I.e., messages that node B and C receive from node A have to be treated differently as messages from D. For our approach we keep the basic idea of the Smart Gossip protocol to establish a parent-sibling-child relationship and derive the gossip probability from this knowledge, but we use a different approach to build up the hierarchy. For being able to differentiate the direction of diffused messages, our approach uses position information.

Additionally, the proposed approach has to deal with varying degrees of mobility. This challenge is not discussed in [9], the authors only consider packet loss due to bad wireless links. This is a second reason why the original Smart Gossip protocol can't be applied in mobile ad hoc networks: the hierarchy is established in a static way and no renewal of neighborhood information is considered.

The main contribution of this work is the design and analysis of a new mechanism for hierarchy creation which considers message directions through position information and treatment of mobility. For this purpose the neighborhood information is not passively collected by overhearing normal gossip messages, but by active exchange of beacon messages. This enables the nodes to receive up to date information of their vicinity in mobile environments. Details of these process are discussed in the next section.

4 Protocol Description

In VANETs the message propagation – as well as node movement – is restricted to streets. Leaving intersections aside, messages can be propagated in two directions: in and against the driving direction. Therefore, e.g. in highway scenarios it is still possible to build a dependency graph even with multiple senders. The protocol has to distinguish between these two possible dissemination directions and build up the hierarchy accordingly. For solving the problem of message propagation direction, information about node positions is used in this work.

The idea is to build the parent-sibling-child relationship in a directed way. I.e. a node can only have parents from one direction. If we resume Figure 2 again, this means, that only node A or D could be a parent of nodes B and C and not both together. Thus, depending on the direction, node A is declared as parent and node D as child or vice versa. Based on the propagation direction of messages this means, that the hierarchy can be built in two ways: in driving direction and against driving direction. If we consider a traffic jam scenario, where vehicles which are approaching the traffic jam have to be informed, it is necessary to build the hierarchy against driving direction. In this case D would be defined as child and A as parent of B and C. In this work the hierarchy is built against the driving direction, since it satisfies our requirements of the highway scenario.

Of course, for some scenarios it would be desirable to send messages in both directions. This is simply achievable if at each node two different neighbor tables are held containing the parent-sibling-child relationships differentiated into the two possible directions.

Listing 1.1. Pseudo code of the proposed protocol

```
Receive_Beacon(fromNode j)
{
  RemoveFromNeighborSets(j);
  if (parent(j) not in NeighborSets &&
        position(j) in driving direction){
    AddToParentSet(j);
  }
  else if (parent(j) not in NeighborSets &&
        position(j) not in driving){
    AddToChildSet(j);
  }
  else if (parent(j) in SiblingSet){
    AddToChildSet(j);
  }
  else if (parent(j) in ParentSet){
    AddToSiblingSet(j);
  }
}
```

As mentioned in the last section another difference to the Smart Gossip protocol is the use of beacons for building the hierarchy. Every vehicle generates beacons and sends them to their 1-hop neighbors. Unlike gossip messages, beacons are not forwarded by the nodes, they are only used to exchange neighborhood information for building up the parent-sibling-child relationship. Additionally, mobility is an important factor. Due to mobility, the neighborship of a node changes frequently. Therefore building the hierarchy must be a continuous process, which is carried out in regular intervals to adapt to topology changes quickly.

To deal with fast changes we added a timestamp based check if entries in the neighborhood table are up-to-date. If a node does not receive a beacon from a neighbor within a certain period it removes the neighbor from its neighbor table. It is obvious, that in this case the hierarchy has to be adapted to the changing neighborship. Therefore, at each reception of a beacon the originator is assigned a role (parent, sibling or child) according to the newest neighborhood information.

The code listing 1.1 shows how the neighbor relationship is established step by step when a node receives a beacon message. At each reception of a packet the method *Received_Beacon* is called. First the sender is removed from the neighbor tables, which are the parent, sibling and child sets. If the parent of the sender (node j in this case) is not in the neighbor tables (thus it is unknown at the

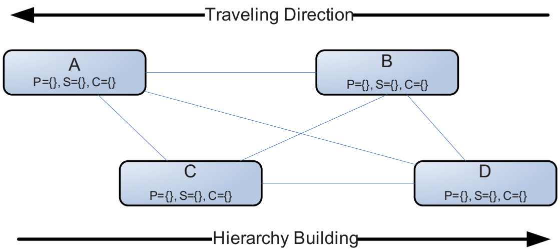


Fig. 3. Example for building the neighbor relationships: initial situation

receiver node) and the sender is in front of the receiving node, the sender is inserted into the parent set. If the parent of the sender is unknown, but the sender is behind of the receiver, the sender is inserted into the child set. When the parent of the sender is known, then depending if it is in the sibling or parent set, the sender is added to the child or sibling set.

In the following an illustrative example for building this parent-sibling-child relationship is given. Figure 3 shows the initial situation. There are 4 vehicles (Nodes A, B, C and D) and they have no information about each other so far. This means, the neighbor tables (parent, sibling and a child set) of those nodes are empty. Since every vehicle sends periodically beacons (including position information and a list with its parents), the hierarchy can be built up step by step. Assume in this example that node D sends first such a beacon which is received from the other three nodes. The nodes A, B and C have to determine their relation to node D. First, the direction to D is checked. Since D is behind the other nodes, it cannot be a parent node. Because the received beacon has an empty list of parents of the sender, D can not be a sibling of the other nodes. Thus, D is put into the child set of vehicles A, B and C.

Continuing the example, assume node B sends the next beacon. The nodes A and C put B into their child list for the same reason as before. Node D determines the direction to node B. Since B is before D, it can not be a child. And because B has no parents determined yet, they can not be siblings. Thus, node D adds B into its parent set. Next, let node A send a beacon. Nodes B, C and D receive the beacon and insert A in their parent set because node A has not the same parents as the other nodes (actually A has no parents at all) and the position of A is in front of B, C and D. In the next step node C sends a beacon. A inserts C in its child set because the parent of C is A itself. Nodes B and D have the same parent as C (node A) and therefore they put C in their sibling set. Figure 4 shows the updated neighbor tables after this step.

In the following a second round is considered where the nodes send in the same order as before. This will show how the topology neighborhood relationship is adapted if newer or more precise information are available. Node D sends a beacon again. For node A there is no change because in the beacon message A is specified as a parent of D and D is already a child of A. But B and C put

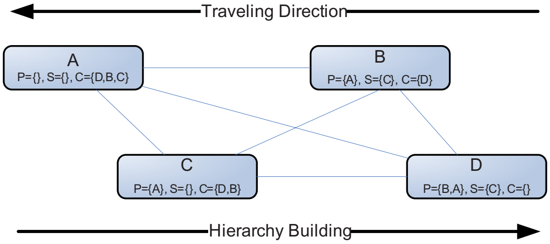


Fig. 4. Example for building the neighbor relationships: situation after all nodes sent 1 beacon

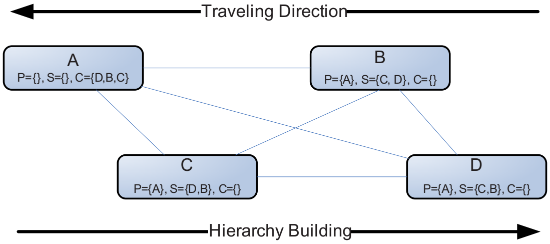


Fig. 5. Example for building the neighbor relationships: the final neighborhood relationships

node D in their sibling set because they have the same parent as D. Next, node B sends a beacon. Again the sets of A are unchanged. Node C and D insert B into their sibling sets cause they have the same parent as B. With further beacons the neighbor tables does not change anymore, since the relationship is already built up correctly. Figure 5 shows the final neighbor tables of these nodes.

Of course, due to mobility the neighbor tables can change again. Nodes can leave the neighborhood of other nodes, or new nodes can arrive but also a shift of the relationship based on the relative positions is possible. But through the use of beacons and timestamps for establishing the relationship, accurate neighbor relationships can be maintained even for highly mobile nodes. An evaluation of the performance for static scenarios as well as for highly mobile nodes is given in the next section.

5 Analysis

To evaluate the performance of the proposed dissemination protocol and to compare it with the Smart Gossip approach we have conducted extensive simulations. For this study we used the Java based network simulator JiST/SWANS [10]. The simulation setups are divided into three parts: in the first setup we use similar simulation parameters as in [9] and make a comparison between our protocol

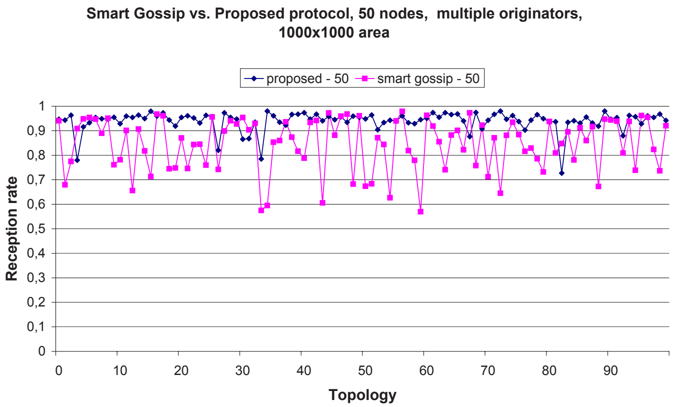


Fig. 6. Performance evaluation of Smart Gossip and proposed protocol with multiple originators

and Smart Gossip with multiple message originators. In the second setup we compare the performance of the two protocols in the highway scenario described in Section 2. We evaluate the performance – reliability and efficiency – of these protocols for varying node densities. In the last setup we investigate the impact of mobility and show the simulation results for different node speeds and densities.

For the first simulation setup, 50 nodes are randomly placed on a field with a size of 1000 per 1000 meters. The wireless transmission range is set to 280 m. Mobility is not considered in this setup, thus nodes are static. The only difference to the parameters used in [9] is the number of message originators in the network. In this setup multiple nodes can initiate broadcast messages. As it can be seen in Figure 6, the delivery ratio of the Smart Gossip protocol drops notably. The fluctuation of the achieved delivery ratio is high and in many cases it drops below 90%. According to the authors from [9] the delivery ratio should be about 99% with one message originator. Thus, multiple message originators have a high impact on the Smart Gossip protocol, in a scenario with low node density as in this case. On the other hand, our proposed protocol achieves much better results and a lower deviation at the same time. For a better comparability we included pure flooding into our simulations. The average delivery ratio for the three broadcasting mechanisms in 100 simulation runs is shown in Table 1. As we can see the best delivery ratio is achieved with flooding. This is obvious, since all nodes retransmit the broadcasted message, achieving a high reliability at the cost of communication complexity. This is the main problem of flooding, especially in dense networks the high number of redundant messages causes channel congestion, resulting in a drastically dropping of the delivery ratio in that case. On the other hand, the delivery ratio between our proposed protocol and flooding differs only by 2%, while the original Smart Gossip ratio is in average almost 10% lower than the proposed protocol.

Table 1. Delivery ratio Flooding, Proposed protocol and Smart Gossip

Flooding	Proposed protocol	Smart Gossip
95.8%	93.9%	84.4%

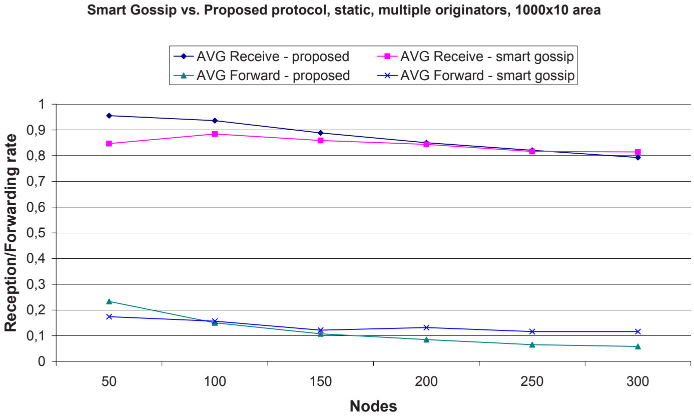


Fig. 7. Performance evaluation of Smart Gossip and proposed protocol with multiple originators on a 1000 per 10 meter field

So far a comparison of the selected broadcast protocols was given in a network topology as used in [9]. Now the focus lies on the highway scenario. Therefore a field of 1000 per 10 meters is used, which should represent a road segment. The other simulation parameters are the same as in the last simulations.

Figure 7 shows the result for varying node densities. As it can be seen, the proposed protocol outperforms Smart Gossip for almost all evaluated node densities in terms of reliability and communication complexity. In the case of 50 nodes the Smart Gossip protocol has a lower average forwarding rate than the proposed. But it should be considered that also the average reception ratio for the Smart Gossip protocol at this node density is approximately 10% lower. Therefore, a new metric is needed that combines these two measured values – reception rate and forwarding rate – and enables thus a better comparison between the protocols. This combined metric can be specified in the following way:

$$Efficiency\ Rate = \frac{Reception\ Rate}{Forwarding\ Rate}$$

The higher the efficiency rate is the better is the performance of a protocol. Table 2 gives an overview of the reception rate, forwarding rate and efficiency rate for both protocols. As this values show, the efficiency rate of Smart Gossip is only at 50 nodes better. But the delivery ratio at this density is approx. by 10% lower. If an application needs a high reception rate for this node density, our protocol is the better choice. All other efficiency ratios show that our protocol

Table 2. Performance comparison between Smart Gossip and proposed protocol

	Smart Gossip			Proposed Protocol		
Nodes	*Reception*	*Forwarding*	*Efficiency*	*Reception*	*Forwarding*	*Efficiency*
50	84.7%	17.4%	4.86	95.5%	23.3%	4.09
100	88.4%	15.6%	5.66	93.6%	15.0%	6.24
150	85.9%	12.2%	7.04	88.8%	10.8%	8.22
200	84.3%	13.2%	6.38	85.0%	8.4%	10.11
250	81.6%	11.6%	7.03	82.1%	6.5%	12.63
300	81.4%	11.6%	7.01	79.3%	5.8%	13.67

outperforms with an increasing node density more and more the Smart Gossip protocol. Thus, this dissemination mechanism is better suited then Smart Gossip in a highway scenario for a wide range of road traffic: it performs well in low densities as well as in traffic jams.

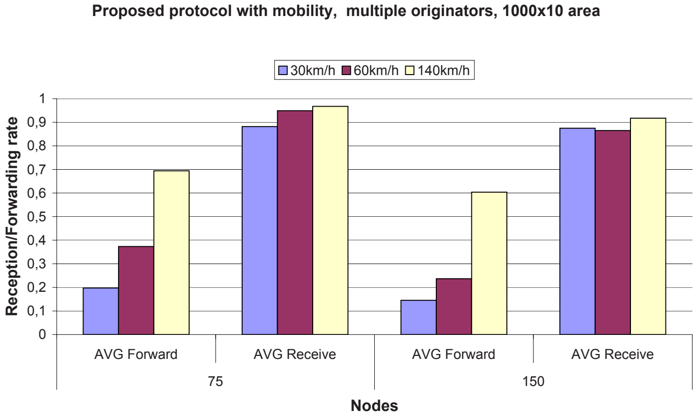


Fig. 8. Impact of mobility on the proposed protocol

In the last simulation setup the performance of the proposed protocol with mobile nodes is investigated. These simulations have been carried out only with our proposed protocol, since the Smart Gossip protocol is not designed to deal with mobility. The neighbor relationship is built up in a static way and no mechanisms were considered to hold such a hierarch up-to-date as it is needed in mobile environments.

For this simulation, also the highway scenario is used, thus nodes are placed into a field with a size of 1000 per 10 meters. For this evaluation the random waypoint mobility model was used, with different node velocities. This mobility model doesn't fit the realistic movements of vehicles on a highway. Nevertheless, this represents a worst-case scenario since nodes are moving in arbitrary directions. A directed movement of nodes into the same direction would better fit to the nature of our protocol where the hierarchy is built depending of road directions. Therefore, if the performance of the protocol is sufficient for this use case, then it should be by far better with a realistic mobility model. As Figure 8 shows, mobility has only a very small impact on the delivery ratio. On the other hand, with higher node velocity the forwarding ratio grows. This is due to the fact, that with higher node speeds the entries in the neighborhood tables are going to be outdated very often. Thus the neighborhood of a node changes often and nodes send with probability 1 if their vicinity is unknown.

It should be noted that in all simulations in this section the delivery ratio was measured by the percentage of reception of a broadcast message at all nodes. This means, a broadcast message is delivered in both directions: into driving direction and against. It is obvious that this situation is not well suited for the proposed protocol. In our approach a directed dependency between neighbors is built, thus a message should be forwarded only against the driving direction. For such a directed forwarding the proposed protocol should achieve a much better performance. We used the general case (actually the worst case) for being able to compare our approach with Smart Gossip in a scenario with multiple message originators.

6 Summary and Outlook

In this work a new probabilistic broadcasting approach derived from the Smart Gossip protocol is introduced. It uses position information to build up neighborhood relationship and – in contrast to Smart Gossip – supports multiple data sources and node mobility. Based on information from beacon messages, appropriate forwarding probabilities can be calculated. This way our approach can adapt well to different network topologies and node densities. The forwarding hierarchy is built up in a directed way. Therefore, the protocol can be applied for networks where nodes have a directed movement, like VANETs for example.

As shown by simulations, the protocol performs well in a wide range of VANET scenarios: it delivers good results in low density networks as well as in networks with high node density. Moreover, node mobility is also considered in this work.

Future work has to show the applicability of the protocol in specific VANET applications: for example a traffic information system which informs upcoming vehicles about traffic density down the road. In such cases, the performance of our protocol should be even better than in the presented simulations, as the messages are forwarded only in one direction and not into the whole network. Other aspects to look at in future work are the investigation of link losses and their impact on the protocol, the evaluation of mechanisms that ensure the robustness of the protocol in such cases, and solutions to deal with road intersections.

References

1. Madden, S., Franklin, M.J., Hellerstein, J.M., Hong, W.: TAG: a Tiny AGgregation service for ad-hoc sensor networks, vol. 36, pp. 131–146. ACM Press, New York, NY, USA (2002)

2. Johnson, D.B., Maltz, D.A.: Dynamic Source Routing in Ad Hoc Wireless Networks. In: Imielinski, T., Korth, H. (eds.) Mobile Computing, vol. 353, pp. 153–181. Kluwer Academic Publishers, Boston, MA (1996)
3. Perkins, C.E., Royer, E.M.: Ad-Hoc On-Demand Distance Vector Routing. In: Proceedings of the 2nd IEEE Workshop on Mobile Computer Systems and Applications, pp. 90–100 (February 1999)
4. Maihöfer, C., Eberhardt, R., Schoch, E.: CGGC: Cached Greedy Geocast. In: Langendoerfer, P., Liu, M., Matta, I., Tsaoussidis, V. (eds.) WWIC. LNCS, vol. 2957, pp. 13–25. Springer, Heidelberg (2004)
5. Ni, S.Y., Tseng, Y.C., Chen, Y.S., Sheu, J.P.: The Broadcast Storm Problem in a Mobile ad hoc Network. In: MOBICOM, pp. 151–162 (1999)
6. Chandra, R., Ramasubramanian, V., Birman, K.: Anonymous Gossip: Improving Multicast Reliability in Mobile Ad-Hoc Networks. Technical report, Ithaca, NY, USA (2001)
7. Haas, Z.J., Halpern, J.Y., Li, L.: Gossip-based ad hoc routing. IEEE/ACM Trans. Netw. 14(3), 479–491 (2006)
8. Miller, M.J., Sengul, C., Gupta, I.: Exploring the Energy-Latency Trade-Off for Broadcasts in Energy-Saving Sensor Networks. In: ICDCS, pp. 17–26 (2005)
9. Kyasanur, P., Choudhury, R.R., Gupta, I.: Smart Gossip: An Adaptive Gossip-based Broadcasting Service for Sensor Networks. In: Mobile Adhoc and Sensor Sysetems (MASS), IEEE International Conference, pp. 91–100 (2006)
10. Barr, R., Haas, Z.J., van Renesse, R.: JiST: an efficient approach to simulation using virtual machines: Research Articles. Softw. Pract. Exper. 35(6), 539–576 (2005)

A Routing Protocol for Balancing Energy Consumption in Heterogeneous Wireless Sensor Networks*

Xiaoya Li, Daoping Huang, and Zonghai Sun

College of Automation Science and Engineering
South China University of Technology , Guangzhou, 510640, China
liyax66@yahoo.com.cn

Abstract. To balance energy consumption for nodes is an important factor considered in wireless sensor networks (WSN). In the paper, we research a heterogeneous wireless sensor networks with two different type sensors which have different initial energy as well as have different length data message to transmit. Each sensor node in such a network is systematically gathering and transmission sensed data to a base station for further processing. We develop and analyze the protocol based on residual energy and energy consumption rate (REECR), which is an energy efficient routing protocol we proposed previously for heterogeneous wireless sensor networks. Although REECR protocol is more energy efficient than Low-Energy Adaptive Clustering Hierarchy (LEACH) protocol, it is not very ideal to balance the energy consumption of nodes, namely, from the first node dies to the last node dies, the time span is long. This paper proposed a zone-based REECR (ZREECR) routing protocol to balance the energy consumption of nodes in the networks, simulation results show that all nodes die from start to end become shorter, the balance of energy consumption between nodes is improved.

Keywords: Wireless sensor networks, balance energy consumption, routing protocol.

1 Introduction

Wireless sensor networks consist of battery-operated sensor devices with computing, data processing, and communicating components. Nodes in the networks sense certain phenomena in the area of interest, and report their observations to a central base station for further analysis. Wireless sensor networks have recently come into prominence because they hold the potential to revolutionize many segments of our economy and life. Wireless sensor networks are suitable for various tasks, including surveillance, widespread environmental monitoring, manufacturing and business asset management, automation in the transportation, security, and health-care industries. Compared to existing infrastructure-based networks, wireless sensor networks can be used in virtually any environment, especially those where wired connections are not possible or the terrain is inhospitable [1, 2].

* This work is supported in part by Nation Science Foundation of Guangdong under grant NO. 06300232.

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 79–88, 2007.

At present, research on wireless sensor networks has generally assumed that nodes are homogeneous. In reality, homogeneous sensor networks hardly exist, even homogeneous sensors also have different capabilities like different levels of initial energy, depletion rate, etc. This leads to the research on heterogeneous networks where at two or more types of nodes are considered. However, most researchers prevalently assume that nodes are divided into two types with different functionalities, powerful nodes and normal nodes. The powerful nodes have more initial energy and fewer amounts than the normal nodes, and they act as clustering-heads as well as relay nodes in heterogeneous networks. Moreover, they all assume the normal nodes have identical length data to transmit to the base station.

In [3], we have researched a heterogeneous sensor networks with two different types of nodes that they have same initial energy but different length data to transmit. Conventional protocols can't ideally adapt the model proposed, so we presented a routing protocol based on residual energy and energy consumption rate (REECR), namely, we elect clustering-heads based on residual energy and energy consumption rate rather than stochastic election and periodical rotation. REECR is more efficient than LEACH (Low-Energy Adaptive Clustering Hierarchy) [4], at the same time, REECR balanced the energy consumption between two different types of nodes better than LEACH, but the REECR protocol is still not perfect to balance the energy consumption in whole networks, it is a little long from the first node dies to the last node dies.

In this paper, we investigate a heterogeneous sensor networks with two different types of nodes which have different initial energy as well as transmit different length data to base station. This network model is a more actual and more complex application network than that in [3]. We further analyze and develop the REECR protocol and make some improvement, and propose zone-based REECR protocol (ZREECR) to better balance the energy consumption in the whole heterogeneous sensor networks. Simulation results approve the protocol we developed.

The remainder of the paper is organized as follows: Section 2 introduces some related works of this paper. Section 3 analyzes the existing problems of REECR protocol. Section 4 provides details about the ZREECR routing algorithm. Section 5 describes the network and radio models. Section 6 provides simulation and analysis of results. Finally, Section 7 summaries our works.

2 Related Works

In homogenous networks, many protocols to balance the energy consumption among nodes have been proposed. One of the most influencing is the LEACH communication protocol. LEACH is a clustering-based protocol employing the approach of randomized rotation of the cluster-heads to evenly distribute the energy load among the sensors in the network, where a small number of clusters are formed in a self-organized manner. A head node in each cluster collects and fuses data from the nodes in its cluster and transmits the result to the base station, LEACH achieves a significant improvement in network lifetime compared with the Direct, MTE (Minimum Transmission Energy), and Static Clustering approach. Based LEACH protocol, more clustered protocols have been proposed, like LEACH-C [5], PEGASIS [6], TEEN [7], BCDCP [8] etc, but they are all under the homogenous condition.

At present, the research to the heterogeneous networks has brought to the attention, and many literatures have obtained some achievements. In [9], authors proposed a probability approach for real-time sensor network applications to assign and optimize sensor systems using heterogeneous functional units with probabilistic execution time. Authors proposed an optimal algorithm to solve the Mode Assignment with Probability (MAP) problem and achieve minimum total cost while the timing constraint is satisfied with a guaranteed confidence probability.

In [10], authors examined the impact of heterogeneous device deployment on lifetime sensing coverage and coverage aging process, and found an optimal heterogeneous deployment can achieve lifetime sensing coverage by several times as much as that with homogeneous deployment considering both initial coverage and the duration of sensing operation as well as the optimum number of high-cost devices in the single-hop communication model that maximizes the lifetime sensing coverage information incorporating several factors that affect the initial sensing coverage and the energy consumption of nodes.

In [11], authors analyzed the operation of a clustered sensor network with two types of nodes, the powerful nodes and the normal nodes. The powerful nodes act as clustering-heads and expend energy much faster than the normal nodes within its cluster until the cluster enters a homogeneous state with all nodes having equal energy levels. Authors examined the time for a cluster to enter this state without any nodes already out of energy, which defines the end of the network's lifetime.

In [12] authors studied the influence of random device deployment on connectivity and lifetime in a large-scale heterogeneous sensor networks and proposed three random deployment strategies for relay nodes in a heterogeneous wireless sensor network, namely, connectivity-oriented, lifetime-oriented and hybrid deployment. Authors investigate how a strategy can affect both connectivity and network lifetime of a multi-hop heterogeneous WSN.

In [13], authors proposed a model that forms clusters of unequal size in the network. Since the powerful sensors are acting as clustering-heads with role of clustering-head is not rotated like in homogeneous network, they have higher work load and thus the network lifetime is bounded by the powerful sensors. By restricting the number of nodes in a cluster based on the energy of the clustering-head, the energy dissipation of the clustering-heads will be balanced.

In [14], authors proposed a topology configuring method for homogeneous wireless sensor networks with an objective of balancing energy consumption over all nodes in the network field without generating any isolated nodes. It was called zone-based method for selecting cluster-heads. Their method starts from dividing a network field into several zones depending on the distance from the origin point. Each sensor node transmits data toward the sink through the nearest neighbor node or cluster-head in each zone. Each cluster-head aggregates data and sends it to the next zone, and continues until all data are transmitted to the sink.

Currently, in heterogeneous networks, most research works generally assume two different types of sensors are deployed with the more powerful sensor having greater processing power and better hardware specifications compared to a normal sensor, see for example [15, 16, 17] etc. However, they hardly research different data transmitting ratio for different type sensor nodes.

In [3], we investigated a heterogeneous sensor network with two different types of nodes possessing same initial energy but sending different length data packet. We found that conventional routing protocols like LEACH etc. can not ideally adapt the network model proposed, therefore, we presented an energy efficient routing protocol-REECR. Where, whether a node becomes a cluster-head for the current round depends on the residual energy and energy consumption rate of the node rather than periodical rotation in turn. The proposed algorithm better balances the energy consumption compared with conventional routing protocols and achieves an obvious improvement on the network lifetime. However, further research found that the REECR protocol still can't very perfectly balance the energy consumption among nodes, so in this paper, we make further research.

3 Problem Formulations

Generally, the energy consumption of clustering-heads is maximal in clustering-based protocols, so the election of clustering-heads is crucial. In the REECR protocol, the election of clustering-heads is stochastic and the number of clustering-heads is determined previously at the first round, and each node has a probability of becoming a cluster-head. Thereafter, the election of clustering-heads is determined in terms of comprehensive consideration of the residual energy and energy consumption rate of nodes. However, looking at a single round of REECR, it is obvious that the cluster-head selection will not automatically lead to balance and minimum energy consumption during data transfer for a given set of nodes. A cluster-head can be located near the edges of the network or adjacent nodes can become cluster-heads. In these cases some nodes have to bridge long distances to reach a cluster-head or the base station. This leads to unbalance and high energy consumption since nodes have to transmit over long distance.

4 Zreecr Protocol

Since the REECR protocol offers no guarantee about the placement of cluster-head nodes, therefore, we would like the cluster-head nodes to be evenly spread throughout the network, this will minimize the distance that the non-cluster-head nodes need to send their data.

One possible solution to this problem is the whole network is divided into several subnets and let clustering-heads separate in location as well as let the total energy of each subnet approximately proportion energy consumption rate respectively, so we propose a zone-based method, which divide a network field into several different size zones depending on the distance and orientation from the BS, as shown in Figure 1. The base station is located in the center of the network field and the zone is configured based on the zone range which is determined by considering the network size, transmission range and orientation on the base station. Considering the clustering-heads closer to the base station need to relay the data from the farther clustering-heads, so we expect that clusters closer to the base station have smaller cluster sizes, thus they will consume lower energy during the intra-cluster data processing, and can preserve some more energy for the inter-cluster relay traffic. Simulation results show that it is the most effective if the area of outside layer subnet divided by that of inside layer subnet is 1.2—1.6. Figure 1 is a schematic diagram with two layer zones, the first layer is divided into 4 equal area zones and the second layer is divided into 8 equal area zones. Each zone area in the first layer is smaller than that in the second layer based on the reason mentioned above. Certainly, the whole network can be divided into three or more layers. Every zone is organized into a static cluster respectively and the clustering-head of each zone takes turns rotation according to rules designed as follows:

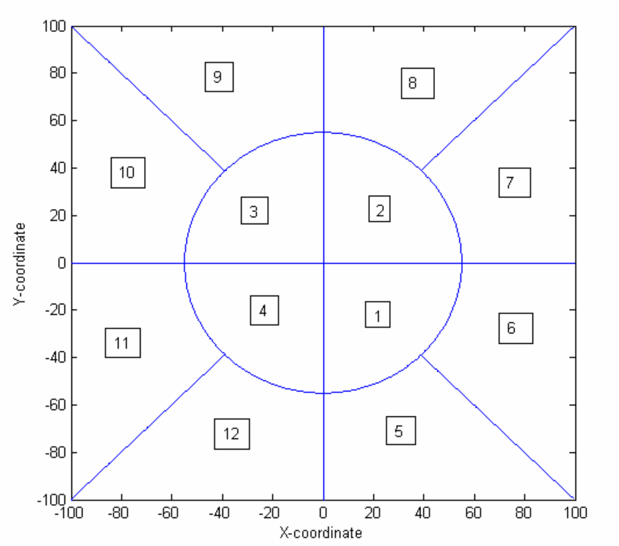


Fig. 1. A schematic diagram of different size zones

At the first round, we choose the nodes that are closest to several specific positions as cluster-heads. These positions are approximately located in geometric center of each zone and almost uniformly distributed in the network avoiding all cluster-heads can be located near the edges of the network or adjacent nodes can become cluster heads. As in the literature [3], from the second round start, the election of clustering-head in each zone based on residual energy and energy consumption rate as follows:

$$P_i(t) = \frac{E_i^{\alpha}(t)}{V_i^{\beta}(t)} . \tag{1}$$

Where $P_i(t)$ is the possibility of each node to be selected as a cluster-head in each zone, α and β are weight coefficients which are decided a prior by experience, $E_i(t)$ is the current residual energy of each node, $V_i(t)$ is the energy consumption rate of each node which is defined as

$$V_i(t) = \frac{E_{initial} - E_i(t)}{r-1}, \quad (r > 1). \tag{2}$$

Where $E_{initial}$ is the initial energy of each node, r is the current rounds.

The data from all sensors in the cluster are collected at the cluster head, which aggregates the data and forwards the aggregated data toward the base station. The forwarding of aggregated packets is done through multiple hops, where every cluster head chooses to forward its data to the closest cluster head in the direction of the base station.

5 Network and Radio Models

We consider a heterogeneous sensor network with two different types of nodes (type-1 and type-0) comprising 200 nodes as shown in Figure 2 and half of them belong to type-1 nodes. The initial energy and transmission data length of the different type nodes are different respectively. The network is running a periodic data collection application.

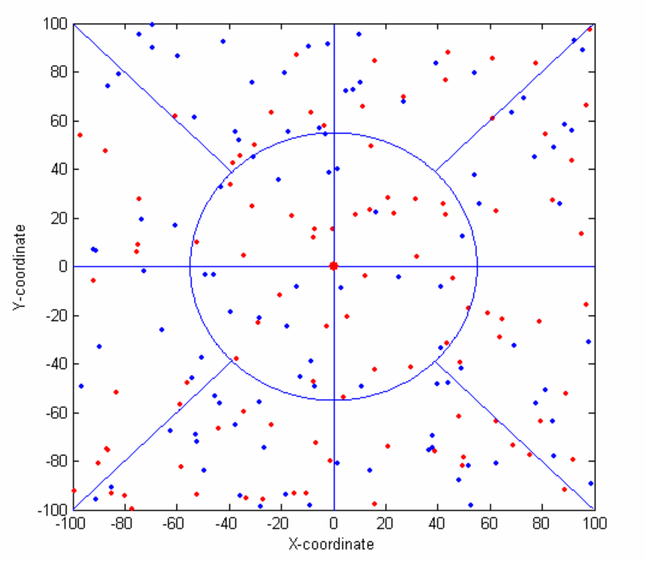


Fig. 2. 200 nodes random network. The blue dots denote type-1 nodes and the red dots denote type-0 nodes.

For our network, it is assumed that

1. A fixed base station is located the center of the sensing field, sensors and the base station are all stationary after deployment.
2. All nodes in the network are energy constrained.
3. All nodes are able to reach the base station.
4. Each node senses the environment at a fixed rate and always has data to transmit to the base station.
5. Communication is symmetric and a sensor can compute the approximate distance based on the received signal strength if the transmission power is given.

For the radio hardware energy dissipation, it is assumed that the transmitter dissipates energy $E_{Tx-elec}(l)$ to run the radio electronics and $E_{Tx-amp}(l,d)$ to run the power amplifier, and the receiver dissipates energy $E_{Rx-elec}(l)$ to run the electronics. Depending on the distance between the transmitter and receiver, both the free space (d^2 power loss) and the multipath fading (d^4 power loss) channel models are used in this paper. Power control is used to invert this loss by appropriately setting the power amplifier, that is, if the distance is less than a threshold d_0, the free space model is used; otherwise the multipath model is used. Thus, to transmit a l-bit message in distance d, the radio expends.

$$\begin{aligned} E_{Tx}(l,d) &= E_{Tx-elec}(l) + E_{Tx-amp}(l,d) \\ &= \begin{cases} lE_{elec} + l\varepsilon_{fs} d^2, d < d_0 \\ lE_{elec} + l\varepsilon_{amp} d^4, d \geq d_0 \end{cases} \end{aligned} \tag{3}$$

and to receive this message, the radio expends

$$E_{Rx}(l) = E_{Rx-elec}(l) = lE_{elec} \tag{4}$$

where E_{elec} is the electronics energy, ε_{fs} and ε_{amp} are the coefficients of the amplifier. The energy spent for aggregating n l-bit messages in a cluster is

$$E_{DA} = nlE_{da} \tag{5}$$

where E_{da} is the energy spent for aggregating l-bit data.

6 Simulation and Results

To evaluate the performances of ZREECR discussed in the previous section, we presented these simulations by MATLAB and compared its performance with other protocols such as LEACH and REECR. The simulation parameters are given in Table 1.

Table 1. Simulation Parameters

Parameter	Value
Network coverage	(-100,-100)～(100,100)m
Base station	(0,0)m
Node number	200
Cluster-head number (LEACH, REECR)	10
Initial energy(type-1)	1.0J
Initial energy(type-0)	0.5J
Eelec	50nJ/bit
ε_{fs}	10pJ/bit/m^2
ε_{amp}	0.0013pJ/bit/m^4
Eda	5nJ/bit/signal
d$_0$	87m
α	5
β	1
Data packet size (type-1)	4000 bits
Data packet size (type-0)	2000 bits

Figure 3 shows the total number of nodes that remain alive over the simulation time. At the aspect of balancing energy consumption for all nodes in the networks, it shows that our method always outperforms LEACH and REECR. That is because the cluster-heads are uniformly distributed over the network to avoid them being located near the edges of the network or too close together. Besides, combine with the

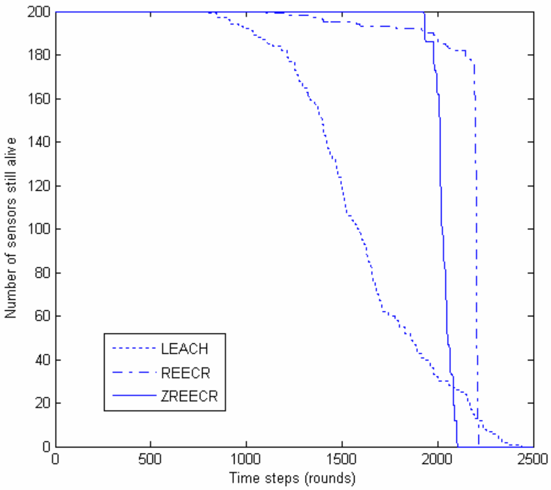


Fig. 3. The simulation results about system lifetime and balance of energy consumption using LEACH, REECR and ZREECR

residual energy and energy consumption rate available in each node, this leads to a balanced energy consumption of all nodes. Therefore, all other nodes in the network die rapidly since the first node dies. Table 2 compares the time span from the first node dies to the last node dies in different routing protocols, it also shows that ZREECR can ideally balance energy consumption of all nodes in the network.

Figure 3 and Table 2 also shows that ZREECR is approximate but not more energy efficient than REECR, that is because we divide the sensing network into several zones and choose the clustering-heads in each zone rather than in the whole networks, so we can't obtain the optimal elections of clustering-heads of the whole network but only obtain local optimization. This implies that we achieve the balance of energy consumption of all nodes at the cost of decreasing a little energy efficiency.

Table 2. Comparison the time span from the first node dies to the last node dies using different protocols

Energy (J/node)	**Protocol**	**Round first node dies**	**Round last node dies**	**Time span**
Type-1 node (1J) Type-0 node (0.5J)	LEACH	841	2412	1571
	REECR	1105	2216	1111
	ZREECR	1907	2104	197

7 Conclusions

In this paper, we propose an approach for the zone-based organization of wireless sensor networks where, in order to balance the energy consumption of all nodes, zone-based clusters are formed. Through analysis and extensive simulations of different routing protocols in heterogeneous wireless sensor networks, we show that our proposed scheme achieves an improvement in balance of the energy consumption compared with the REECR protocol we proposed previously and LEACH. Our results show that this direction has the potential to improve performance in terms of the balance of energy consumption of all nodes in networks. However, on the other hand, the results also show that the energy efficiency of our scheme has a bit decreased compared with the REECR, therefore, the ZREECR protocol need to further improve in the future.

References

[1] Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: A Survey on Sensor Networks. IEEE Communications Magazine 40(8), 102–114 (2002)

[2] Al-Karaki, J.N., Kamal, A.E.: Routing techniques in wireless sensor networks: a survey. IEEE Wireless Communications 11(6), 6–28 (2004)

[3] Li, X., Huang, D., Yang, J.: Energy Efficient Routing Protocol Based on Residual Energy and Energy Consumption Rate for Heterogeneous Wireless Sensor Networks. In: The 26th Chinese Control Conference, vol. 5, pp. 587–590 (2007)

[4] Heinzelman, W.R., Chandrakasan, A.P., Balakrishnan, H.: Energy-Efficient Communication Protocol for Wireless Microsensor Networks. In: Proc. 33rd Hawaii Inter. Conf. on System Science, pp. 10–20 (2000)

[5] Heinzelman, W.B., Chandrakasan, A.P., Balakrishnan, H.: An Application-Specific Protocol Architecture for Wireless Microsensor Networks. IEEE Trans. Wireless Commun. 1(4), 660–670 (2002)

[6] Lindsey, S., Raghavendra, C.S.: PEGASIS: Power-efficient gathering in sensor information systems. In: Proc. of the IEEE Aerospace Conf. Montana: IEEE Aerospace and Electronic Systems Society, pp. 1125–1130 (2002)

[7] Manjeshwar, A., Agrawal, D.P.: TEEN: A protocol for enhanced efficiency in wireless sensor networks. In: Int'l Proc. of the 15th Parallel and Distributed Processing Symp., pp. 2009–2015. IEEE Computer Society, San Francisco (2001)

[8] Muruganthan, S.D, Ma, D.C.F., Bhasin, R.I.: A Centralized Energy-Efficient Routing Protocol for Wireless Sensor Networks. IEEE Communications Magazine 43(3), 8–13 (2005)

[9] Qiu, M., Xue, C., Shao, Z., Zhuge, Q., Liu, M., Sha Edwin, H.M.: Efficent Algorithm of Energy Minimization for Heterogeneous Wireless Sensor Network. In: Sha, E., Han, S.-K., Xu, C.-Z., Kim, M.H., Yang, L.T., Xiao, B. (eds.) EUC 2006. LNCS, vol. 4096, pp. 25–34. Springer, Heidelberg (2006)

[10] Jae-Joon, L., Bhaskar, K., Kuo, C.C.: Impact of Heterogeneous Deployment on Lifetime Sensing Coverage in Sensor Networks. In: First Annual IEEE Communications Society Conference on Senor and Ad Hoc Communications and Networks, pp. 367–376 (2004)

[11] Lee, H.Y., Seah, W.K.G., Sun, P.: Energy Implications of Clustering in Heterogeneous Wireless Sensor Networks- An Analytical View. In: The 17th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, pp. 1–5 (2006)

[12] Kenan, X., Hossam, H., Glen, T.: Relay Node Deployment Strategies in Heterogeneous Wireless Sensor Networks:Multiple-Hop Communication Case. In: Second Annual IEEE Communications Society Conference on Senor and Ad Hoc Communications and Networks, pp. 575–585 (2005)

[13] Soro, S., Heinzelman, W.B.: Prolonging the Lifetime of Wireless Sensor Networks via Unequal Clustering. In: Proc. IEEE Inter. Conf. on Parallel and Distributed Processing Symposium, pp. 4–8 (2005)

[14] Kim, K., Kim, H., Han, K.: A Zone-Based Method for Selecting Clusterheads in Wireless Sensor Networks. In: RSCTC. The Fifth International Conference on Rough Sets and Current Trends in Computing, Kobe, Japan, pp. 667–676 (November 2006)

[15] Mhatre, V.P., Rosenberg, C., Kofman, D., Shroff, N.: A Minimum Cost Heterogeneous Sensor Network with a Lifetime Constraint. IEEE Transactions on Mobile Computing 4(1) (2005)

[16] Duarte-Melo, E.J., Liu, M.: Analysis of Energy Consumption and Lifetime of Heterogeneous Wireless Sensor Networks. In: Proceedings of IEEE Globecom, Taipei, Taiwan (2002)

[17] Qing, L., Qingxin, Z., Mingwen, W.: A Distributed Energy-Efficient Clustering Algorithm for Heterogeneous Wireless Sensor Networks. Journal of Software 17(3), 481–489 (2006) (in Chinese with English abstract)

Novel Hierarchical Based Route Optimization Scheme for Nested Mobile Network: Design and Performance Evaluation*

Xiaohua Chen, Yajuan Qin, Ping Dong, and Zhiwei Yan

Beijing Jiaotong University, 100044 Beijing, China
05111040@bjtu.edu.cn

Abstract. To alleviate the problem of sub-optimal routing and multiple encapsulations in nested mobile network, a hierarchical based route optimization scheme (HBRO) is proposed. Root mobile router (MR) works as mobility anchor point (MAP) and transmits packets for the mobile network nodes (MNNs). Following advantages can be achieved with this scheme: 1) correspondent node (CN) transmits packet directly to MAP, bypassing all of the home agents (HA) of MRs; 2) tunnel exists only inside the mobile network. We build an analysis model for for performance evaluation, and the analysis results indicate that packet delivery overhead decreases significantly, and the delivery efficiency is beyond 80 percent when packet size is larger than 200 bytes and network resources are utilized to a great extent.

1 Introduction

Network mobility (NEMO) protocol [1] provides the mobility of an entire network which changes its point of attachment to the Internet. Based on MIPv6 [2], this protocol uses bidirectional tunnel between MR and HA to provide connectivity for MNNs. There are mainly two kinds of MNN: local fixed node (LFN) and visiting mobile node (VMN). LFN belongs to the subnet of an MR and has no mobility support; while VMN is temporarily attached to the MR's subnet by obtaining its care of address (CoA). Multiple mobile networks can be nested in a hierarchical form, which is called nested mobile network.

The architecture of nested NEMO is shown in Figure 1. The whole network moves away from the home network and attaches to the foreign network through access router (AR). CN in foreign network establishes connection with LFN. According to NEMO basic support protocol, packets from CN to LFN should traverse HA of MR4, HA of MR2, HA of MR1, MR1, MR2 and MR4, as shown in Figure.1, which is redundant and belongs to sub-optimal routing. Furthermore, because LFN attaches to MR4, which is in 3 level of nest, packets between MR1 and HA of MR1 are encapsulated 3 times, so if the foreign network is

* This research is supported by the National Science Foundation of China (No. 60573001, 60674008) and the China Next Generation Internet Project [2006-2075].

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 89–100, 2007.

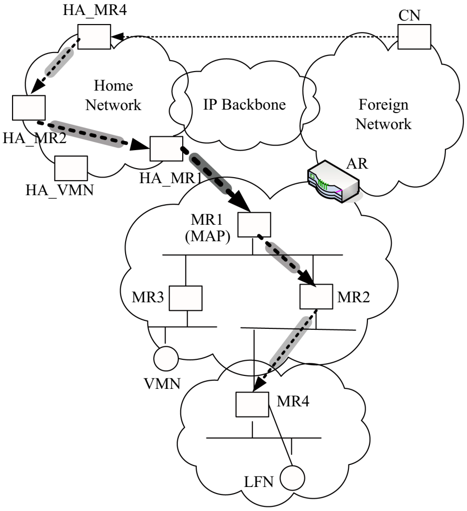


Fig. 1. Architecture of Nested mobile network and sub-optimal routing between CN and LFN

far away from the home network, packets must traverse long distance through the core network, which will inevitably increase the packet delivery delay and overhead, and it will get worse with the increase of nest level. In order to alleviate sub-optimal routing and multiple encapsulations in NEMO, several route optimization (RO) solutions have been proposed [4-12]. According to different requirements and scenarios, these solutions can be classified into three categories: 1) route optimization without the participation of nodes inside the network, such as PSBU proposed by Ernst et al [5]; 2) route optimization with the participation of nodes inside the network, such as prefix delegation scheme proposed by Kniveton et al. [6]; 3) proposals specifically addressing the nested mobility issue, such as reverse routing header (RRH) scheme proposed by Thubert et al [7].

RRH scheme [7] is the most referred one and it uses a new type routing header called RRH to record the route out of the nested mobile network and converted it to type 2 routing header for packets destined for mobile network. Such a scheme can alleviate the sub-optimal routing and multiple encapsulations problem, while it has the following challenges: 1) the routing header contained in every packet between root-MR and HA introduces high overhead, especially when the mobile network is deep nested; 2) sub-optimal routing still exists, because all packets between CN and MNN should first be destined for HA of sub-MR. So when the mobile network moves away from the home network, tunnel between sub-MR and HA will cause much packet delivery delay.

In this paper we propose a novel route optimization scheme based on HMIPv6 [3]. This scheme introduces the idea of hierarchical management into nested mobile network to reduce packet delivery overhead and signaling message number. We focus on the solution for LFN, which can be extended to VMN easily.

The rest of this paper is organized as follows: in section 2, HMIPv6 is presented as related work. The HBRO scheme is proposed in section 3, and in section 4 the evaluation model is built and in section 5 the network topology is presented and performance of HBRO is analyzed. Finally in section 6 we conclude the whole work.

2 Related Work

In MIPv6, if mobile node (MN) moves from an AR to another, it configures CoA and sends binding update (BU) to HA. HA creates binding item for MN (note that, mobile node in mobile network is called MNN, mobile node in MIPv6 is called MN), then packets for MN are transmitted by HA. This method will induce delivery latency, so MIPv6 provides route optimization support: MN performs return routability (RR) procedure with CN, such a procedure involves that MN sends home test init (HoTI) and care of test init (CoTI), receives home test (HoT) and care of test (CoT), and CN creates a binding item of MN's HoA and CoA, then CN can communicate directly with MN and the delivery latency is reduced.

HMIPv6 reduces the signaling load and registration latency by introducing an entity of mobility anchor point (MAP). Mobility management inside the local domain is handled by MAP, and mobility between different MAPs is handled by HA. in this protocol MAP basically acts as a local HA in the foreign network, tunneling packets to current location of MN.The operation flow sequence of HMIPv6 is shown in Figure.2.

When MN enters a new MAP domain, it will receive router advertisement (RA) containing information of one or more local MAPs. Despite on-link CoA (LCoA) formed normally, MN forms regional CoA (RCoA) with the new MAP option and RCoA is registered with HA and CNs. When MN changes its point of attachment inside the MAP domain, it only needs to register the new LCoA with MAP. MAP intercepts the packets heading for the old LCoA and tunnels them to the new LCoA just like HA.

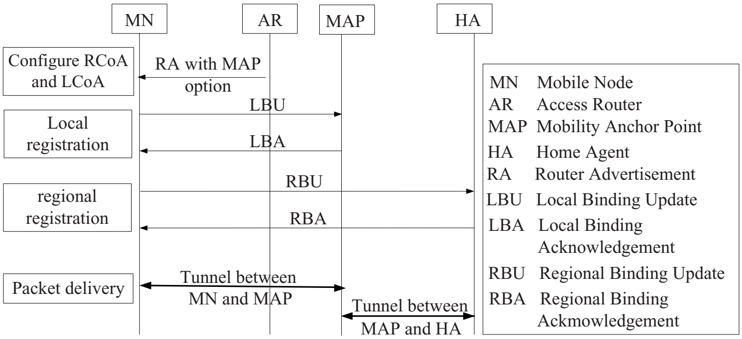


Fig. 2. Operation flow sequence of Hierarchical MIPv6

HMIPv6 is designed to reduce the amount of signaling between MN, CN, and HA, it can also improve MN's handover performance. The characteristics of hierarchical management in HMIPv6 can be used in nested NEMO to alleviate the sub-optimal routing and multiple encapsulations caused by deep nest. However, in HMIPv6 only MN changes its address, while in NEMO, both MN and MR change addresses frequently, so modifications are required on existing HMIPv6 protocol.

3 Proposed Route Optimization Scheme

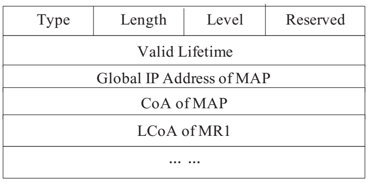
MR has one or more ingress interfaces and egress interfaces. When more egress interface exists, it will cause multihoming related issues [13], which is not in the scope of this paper. Egress interface receives traffic from Internet, while ingress interface advertises mobile network prefix (MNP) and provides service for MNNs.

When MR moves to foreign network, it will receive RA and configures CoA. If there is no MAP option in RA message, MR finds itself top level MR, then it works as MAP, creates MAP option in RA and adds its new CoA in MAP option. If MAP option exists, MR configures not only RCoA based on prefix of MAP's global address, it also configures LCoA on access MR's MNP. MR adds its LCoA in the RA, and advertises it out. In our protocol, MAP option is extended to record all LCoAs of downstream MRs along the nest path. This option is shown in Figure.3.

Route optimization procedure: MR performs RR procedure on behalf of LFN because LFN has no mobility support. Take the RO procedure between CN and LFN in Figure.1 as example, when MR4 receives packets tunneled from HA of MR4 and delivers it to LFN, RO procedure is triggered as shown in Figure.4.

Optimized path: after the routing path is established, packets from CN are transmitted directly to CoA of MAP; MAP checks the type 2 routing header, finding that the packets are destined for LFN, then it checks its binding cache for LFN and builds tunnel to MR4 and MR4 delivers the packets to LFN. Reversely, packets from LFN using MAP's CoA as the source address, CN as destination; in outer header the source address is LFN and destination address is MAP's CoA. When packets arrive, MAP decapsulates and send them to CN.

Mobility management: with the hierarchical management, when MR changes its current address within the MAP domain, it only needs to register the new



Type	Length	Level	Reserved
Valid Lifetime			
Global IP Address of MAP			
CoA of MAP			
LCoA of MR1			
... ...			

Fig. 3. Extended MAP option

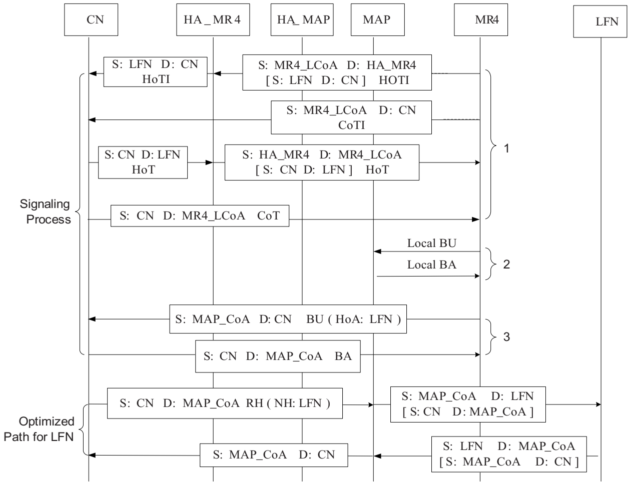


Fig. 4. Packet flow sequence in route optimization procedure for CN and LFN

LCoA with MAP, without triggering any update to HA, so signaling load and registration latency is reduced. If MAP handovers and gets a new CoA, it will send BU to all CNs for its MNNs. Its new CoA is also included in RA, so all MRs and MNNs can use this new CoA as the source address in outbound data packets.

Extension to VMN: this solution can be extended to VMN easily because VMN has the mobility support and it can perform all the route optimization procedure described above.

4 Performance Evaluation

In this section, we will describe the development of analytical model to evaluate the performance of packet delivery overheads, delay and efficiency, which are directly related to the objectives of route optimization.

4.1 Analysis of Packet Delivery Overheads

Packet delivery overhead is the additional cumulative traffic load on the network induced by mobility management functions. It includes header overhead and signaling overhead. Header overhead is caused by additional information in packet headers, such as encapsulation, RH type0 and RH type2 required for routing packets from source to destination, basic IPv6 header is not included. Signaling overhead is caused by signaling required for performing RO.

As described above, NEMO basic support protocol and RRH have only header overhead, while HBRO has not only header overhead, it has also signaling overhead.

Table 1. Additional header size of different schemes

Scheme(s)	p_i^+	conditions(n)
Basic	$h \times i$	$1 \leq i \leq n$
	$h \times (2n - i)$	$n \leq i \leq 2n$
RRH	$h + [8 + (a \times n)]$	$i = 1$
	$8 + (a \times n)$	$2 \leq i \leq n$
HBRO	$8 + a$	$i = 0$
	$h + [8 + a]$	$1 \leq i \leq n$

As defined in [14], header overhead is the product of the additional header size and the distance it travels in the networks. The per packet header overhead of an RO scheme s in a mobile network with n levels of nested mobility is given as:

$$H_o(s, n) = \sum_R (p_i^+ \times d_i) \qquad (1)$$

Where R is the ordered set of nodes along the data delivery path from CN to MN, MN is excluded. d_i is the distance in terms of IP hops between node i to i+1, and (p_i^+ is the additional header size (excluding the IPv6 header) that is transported over the distance d_i. (p_i^+ for different schemes can be expressed in Table.1.

In this table, h and a are IPv6 header size (40 bytes) and IPv6 address size (16 bytes), respectively. Signaling messages required for RO are carried in the mobility header, which is an extension header defined in MIPv6. Based on the specification, the sizes of mobility header containing different messages are listed in Table.2.

Router advertisement(+) means the additional bytes of extended RA based on MIPv6 specification. The RA message used in MIPv6 is 56 Bytes, while in HBRO, RA sent by MAP is 104 Bytes. Furthermore, when the nest level increases

Table 2. RO messages and size (IPv6 header is not included, i represents nest level)

Type	Size (Bytes)
Binding Update	80
Router Advertisement(+)	$48 + (i - 1) \times a$
Binding Acknowledgement	40
Home Test init	16
Care of Test init	16
Home Test	24
Care of Test	24
Local BU	$80 + (i - 1) \times a$
Local BA	40

by 1, MR will add its LCoA in the RA so an address of 16 Bytes is added. The signaling overhead is shown as below:

$$S_o(s,n) = \sum_R \sum_Q (sp_i \times d_i) \tag{2}$$

Where sp_i is the size of a signal packet including one or more IPv6 headers that travel the distance of d_i, and Q is the set of messages required to complete the RO process. Header overhead is present in every data packet, but signaling overhead is induced only once when performing RO. Therefore the average signaling overhead per packet can be calculated by dividing the signaling overhead by the number of data packets transmitted over the optimized path, supposed E, then the average overhead per packet is given below:

$$T_o(s,n) = H_o(s,n) + \frac{S_o(s,n)}{E} \tag{3}$$

4.2 Analysis of Packet Delivery Delays

Packet delivery delay refers to the time required to transmit a packet from source (CN) to destination (MN). Let $t(p_i)$ denote the delivery delay of a packet of size $p_i(p_i = p + p_i^+)$ from the ith node to the $(i+1)th$ nodes along the optimized path, then:

$$t(p_i) = t_p + [t_r + \frac{p_i}{B_w} + l_w] \times d_i \tag{4}$$

t_p is the additional processing delay at a node, such as binding cache lookup and encapsulation, header modification. Assume that a CN generates data packets destined to the MN at a mean rate λ, and the MN moves from one subnet to another at a mean rate μ. We define packet to mobility ratio (PMR,ρ) as the mean number of packets received by the MN from the CN per movement. When the movement and packet generation processes are independent and stationary, the PMR is given by $\rho = \lambda/\mu$. t_p can be calculated as below:

$$t_p = \frac{\rho}{\lambda(1-\rho)} = \frac{1}{\mu(1-\rho)} \tag{5}$$

Then the overall packet delivery delay from CN to MN in scheme s with n levels of nest is given as the sum of delays on wire and wireless links:

$$T(s,n) = t_p + [t_r + \frac{p_i}{B_w} + l_w] \times d_i + t_p + [t_r + \frac{p_i}{B_{wl}} + l_{wl}] \times d_i \tag{6}$$

Parameters for packet delivery delay analysis are taken from paper by Lo et al. [14].

Table 3. Parameters for packet delivery delay analysis

B_w	B_{wl}	L_w	L_{wl}	P	$1/\mu$	t_r
100Mbps	11Mbps(+)	0.5ms	2ms	1000Bytes	4ms	0.001ms

4.3 Analysis of Packet Delivery Efficiency

Packet delivery efficiency measures how effectively the network resources are utilized. The efficiency is calculated by dividing the total traffic load when an IP packet is routed normally (without using mobility support) by the traffic load when the same packet is routed via routing path established by different routing scheme. That is, when a packet is routed normally, there is no associated overhead as long as the packet is routed through the shortest path as determined by routing algorithm. So the delivery efficiency of route optimization scheme s with nest level of n can be presented as:

$$E(s,n) = \frac{P \times d_s}{\sum(p \times d_i) + T_o(s,n)} \tag{7}$$

Where P is the size of an IP packet that includes a basic IPv6 header and payload, d_s is the shortest path between source and destination.

5 Analysis and Simulation Results

Figure.5 gives a generalized description of a network configuration which can be extended to any scenario. The numbers represent distance in terms of IP hops between two nodes.

5.1 Delivery Overhead

Delivery overhead includes header overhead and signaling overhead. Figure.6 shows the analysis of delivery overhead with respect to the average number of packets transported over the optimal path. NEMO basic support protocol and RRH scheme have no signaling message, in this figure their overhead will not

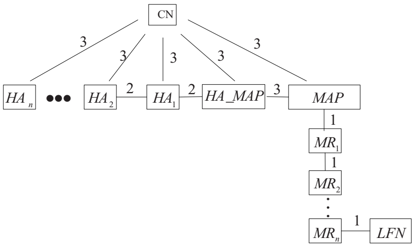


Fig. 5. Relative distances in hops in the simulated network topology

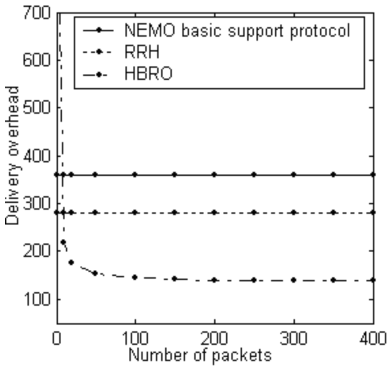


Fig. 6. Delivery overhead VS number of packets transported (n=2)

vary with the change of number of packets transmitted. HBRO has relatively larger overheads when only a few packets are transported because a number of signaling messages are required to complete the RO process. However, as the number of packets increases, the overhead decreases abruptly. So for a larger data transfer session, it is desirable to use HBRO scheme.

Figure.7 gives the delivery overhead comparison of NEMO basic support protocol and HBRO scheme with different nested level.

From Figure.7 we can see that the packet delivery overhead is largely reduced by the use of HBRO scheme when there are more levels of nested mobility. For instance, HBRO reduces the per-packet overhead by 43.56 percent and 81.59 percent for E=1 and E=200 respectively when the level of nest is 5.

5.2 Delivery Delay

According to analysis in section 4.2, processing time at every node contributes greatly to the total delivery latency, such as binding cache lookup, encapsulation /decapsulation etc. In NEMO basic support protocol, n MRs, n HAs, CN and

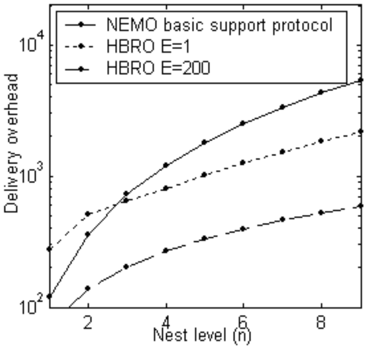


Fig. 7. Delivery overhead VS nested level

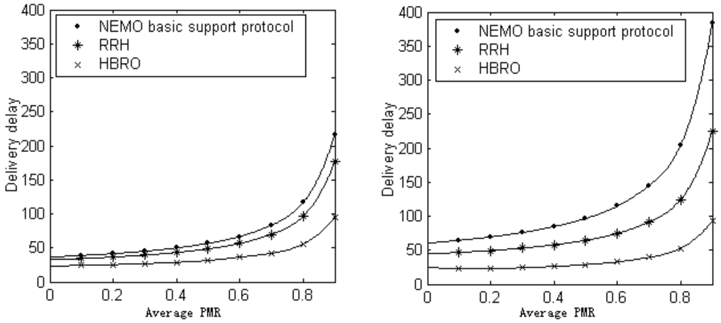


Fig. 8. Comparison of delivery delay(n=2 and n=4)

MN totally $2n + 2$ nodes are included in the packet processing, Similarly, in RRH scheme, $n + 3$ nodes participate in packet processing, they are CN, MN, sub-MR and its HA. While in HBRO scheme, CN, MAP, proxy MR and LFN are involved, independent of the nest level. Other MRs along the routing path only performs routing table lookup, consuming about 0.001 ms, which is rather small and can be omitted.

Figure.8 compares the analysis of packet delivery delay for different average PMR on nodes when nest level is 2 and 4, respectively. PMR represent burden on nodes. As expected, the maximum delay occurred in the basic support protocol and the minimum in the HBRO.

From Figure.8 We can see that the delays increased rapidly with increasing of nest levels except HBRO scheme.

5.3 Delivery Efficiency

Figure.9 compares the packet delivery efficiency of different schemes against packet size for n=2. Figure.9 illustrates that the efficiency of NEMO basic

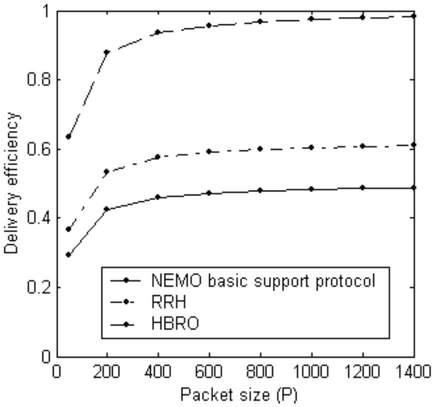


Fig. 9. Packet delivery efficiency

support protocol is very low (about 40 percent to 50 percent) because the suboptimal routing and multiple encapsulations cost much network resources; whereas the delivery efficiency of RRH scheme is a little better and its delivery efficiency is about 50 percent to 60 percent. However, HBRO has the efficiency of more than 80 percent when the packet size is bigger than 200 bytes, furthermore, when the packet size is big enough the efficiency is approximately 100 percent. This is because when the packet size is small, the header overhead and signaling overhead caused by the establishment of optimized path cost some network resource. While when packet size is rather big, signaling overhead becomes relatively small. Packets are transmitted by the most effective manner, and the network resources are utilized to the greatest extent.

6 Conclusions

NEMO basic support protocol provides the mobility management of a whole network, however, is has the side effect of increasing packet delivery overheads due to sub-optimal routing and multiple encapsulations of data packets, which will become worse when the network is nested deeply. To solve this problem, we propose a hierarchical based route optimization scheme HBRO for nested NEMO. This scheme enables CN to forward packets directly to the MAP without tunneling, and tunnel exists only inside the mobile network, which will reduce packet delays and encapsulation overheads between home network and foreign network. Analysis results show that this scheme can improve the system performance, and packet delivery efficiency is increased significantly, which is instructive to the real deployment of mobile network.

References

1. Devarapalli, V., Wakikawa, R., Petrescu, A., Thubert, P.: Network Mobility (NEMO) Basic Support Protocol, RFC 3963 (2005)
2. Johnson, D., Perkins, C., Arkko, J.: Mobility Support in IPv6, RFC 3775 (2004)
3. Soliman, H., Flarion, C., Castelluccia, K., et al.: Hierarchical Mobile IPv6 Mobility Management (HMIPv6), RFC4140 (2005)
4. Perera, E., Sivaraman, V., Seneviratne, A.: Survey on Network Mobility Support. Mobile Computing and Communications Review 8(2) (2005)
5. Ernst, T.: Mobile Networks Support in Mobile IPv6 (Prefix Scope Binding Updates), Draft-ernst-mobileipv6-network-04 (2002)
6. Kniveton: Mobile Network Prefix Delegation, draft-ietf-nemo-prefix-delegation-02, work in progress (2007)
7. Thubert, P., Molteni, M.: IPv6 reverse routing header and its application to mobile networks, draft-thubert-nemo-reverse-routing-header-07 (2007)
8. Ohnishi, H., Sakitani, K., Takagi, Y.: HMIP based route optimization method in a mobile network, draft-ohnishi-nemo-ro-hmip-04 (2003)
9. Cho, H., Paik, E.K., Choi, Y.: RBU: Recursive Binding Update for Route Optimization in Nested Mobile Networks IEEE VTC (2003)

10. Kafle, V.P., Kamioka, E., Yamada, S.: MoRaRo: mobile router-assisted route optimization for network mobility (NEMO) support. IEICE Transaction on Information and Systems E89-D(1), 158–170 (2006)
11. Cho, H., Kwon, T., Choi, Y.: Route Optimization Using Tree Information Option for Nested Mobile Networks. IEEE Journal on Selected Areas in Communications 24(9) (2006)
12. Calderon, M., Carlos, J., Bernardos, Bagnulo, M., et al.: Design and Experimental Evaluation of a Route Optimization Solution for NEMO. IEEE Journal on Selected Areas in Communications 24(9) (2006)
13. Arkko, J.: Issue List for NEMO Multihoming Problem Statement, draft-ietf-nemo-multihoming-issues-07 (2007)
14. Lo, S.-C., Lee, G., Chen, W.-T., Liu, J.-C: Architecture for mobility and QoS support in all-IP wireless networks. IEEE Journal on Selected Areas in Communications 22(4), 691–705 (2004)

Reputation-Based Routing in Hybrid Ad Hoc Networks

Benjamin Tourolle, Sylvie Laniepce, and Mohammed Achemlal

Orange Labs, France Telecom, 42 rue des coutures,
BP 6243, 14 066 CAEN cedex 04, France
{s.laniepce, mohammed.achemlal}@orange-ftgroup.com

Abstract. Within Ad hoc networks, IP connectivity is achieved with the help of multi-hop routing; nodes are supposed to cooperate and act as routers for the benefit of the others. However, this cooperation-based paradigm has limitations, because nodes do not always make their resources available, properly. We propose a reputation-based routing protocol, called MOOR, for application to hybrid Ad hoc networks. MOOR aims at improving the reliability of communications between Ad hoc nodes and the gateway to an infrastructure-based network. We present an application–motivated performance analysis of our protocol MOOR, which allows to quantify the benefit in terms of radio network coverage increase. The impact of non-cooperative nodes on a MOOR protected network remains rather low, even for nodes located 5 hops away from the gateway.

Keywords: Hybrid Ad Hoc Networks, Multipath Routing, End-to-end Reputation.

1 Introduction

A Mobile Ad hoc Network (MANET) is a wireless network composed of autonomous entities – nodes – operating in a self-organized way, without relying on any established infrastructure or centralized administration. Since all network services are assumed to be provided by the nodes themselves, the functioning of Manets depends largely on the cooperative behavior of the nodes. Communications between two distant nodes require intermediate nodes that agree to share their resources for the benefit of the others.

We consider hybrid Ad hoc networks, defined as Ad hoc networks extended by an operator using an infrastructure-based network (in order to provide an access to Internet to the Ad hoc nodes for example). A node that is beyond the direct reach of the operator gateway uses Ad hoc routing mechanisms – that is multi-hop routing – to communicate with its access provider. This new paradigm is of interest for nodes which get Internet connectivity and for the operator who extends its infrastructure-based network coverage at low cost. However, the proper operation of this last application requires the mobile nodes to collaborate with each other although no trust relationship exists between them.

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 101–112, 2007.

Several complementary mechanisms have been proposed to address the non-cooperation issue in pure Ad hoc networks (not interconnected to an infrastructure-based network): incentive schemes that reward cooperative nodes [1] and reputation mechanisms.

Hybrid Manets are of interest to more and more researchers investigating the opportunities for the infrastructure part of the network, to offer services that could enhance the performances of the Ad hoc part [2]. However, to our knowledge, no such study has specifically focused on reputation mechanisms. This paper introduces a new mechanism that exploits the specificities of hybrid Ad hoc networks to address some major non-cooperation issues.

The remainder of this paper is organized as follows: related work is discussed in part II, followed by an overview of the protocol scheme in part III. Part IV gives a technical description of the MOOR protocol before evaluation of its performances in part V. Future work is outlined in part VI and conclusions are drawn in part VII.

2 Related Work

Routing in Manets – Reputation

Traditionally, the non-cooperation issue in Manets is addressed with reputation systems based on a watchdog component operating locally on each node [3]. This consists for each node of overhearing neighbor nodes, in the so-called "promiscuous" mode, to ensure the correct forwarding operation. These mechanisms present several limitations [4]. In particular, reputation values are shared among the nodes through exchanges of recommendations. This implies to deal with a trust issue (false accusation) between the nodes which is at least as complex as the reputation issue [5, 6, 7]. We take the advantage of the relationship between the operator and its customers to address the trust issue in hybrid Ad hoc networks.

End-to-end reputation

The traditional watchdog related proposals implement local interpretation and decision. For this reason, the nodes' knowledge is not always consistent in the network, it may imply sub efficient choices. Instead of using local and hop-by-hop treatment, we implement an alternative end-to-end monitoring technique. This latter one is based on transport layer observations that are used at the routing layer, with the help of cross layer communication. This technique was first introduced by [8] in Manets but reveals its true potential through our proposed implementation in hybrid Manets. With hybrid Manets, we benefit from the presence of a compulsory transit point for all communications between the Internet and the Manet.

3 The MOOR Scheme

3.1 Objectives

Our goal is to increase the reliability of communications between nodes and their access gateway (the network operator). This is particularly crucial for the nodes which are several hops away from the gateway – more than 3 hops – because in this case they are more likely to meet a larger number of non-cooperative nodes. We propose an appropriate routing protocol called MOOR – Multipath Optimized Operator Routing – to address this issue.

We assume that solutions such as the multi-channel one will be able to address the bandwidth degradation over hops observed in conventional Ad hoc networks [9]. We fixed the higher hop count limit between any node and the gateway to 5, which we consider a reasonable value in relation to the network operations (throughput, delay, link failure probability, etc.).

3.2 Overall System Description

MOOR (Figure 1) is a multipath routing protocol able to discover and evaluate several routes between any node and a service gateway, in such a way that the intermediate node contribution is reduced in favor of the destination - the trusted gateway operator - in evaluating the network cooperation. This is different from traditional watchdog techniques where each node makes use of a cooperation evaluation limited to one-hop nodes, assigning these untrusted one-hop nodes the task of performing similarly.

The protocol is designed to limit the impact of non-cooperative forwarding behavior, by recommending routes including intermediate nodes which have revealed the strongest cooperation ability in the past (these routes have the greatest reliability metric).The gateway monitors the communications between the different sources and itself. We can thus evaluate the cooperation of all relay nodes involved in the routing and not only the cooperation of the node neighbors as with a traditional Watchdog component. In other words, node cooperation in the network is monitored on the gateway in order to compute some end-to-end route metrics.

For fully malicious nodes which drop all data packets, no observation occurs at the gateway because no packet reaches the destination. As a consequence of the observations performed on the destination side, initial node reputations must be set very low – nodes are not trusted by default – and are increased when they cooperate: nodes need to demonstrate that they are cooperative.

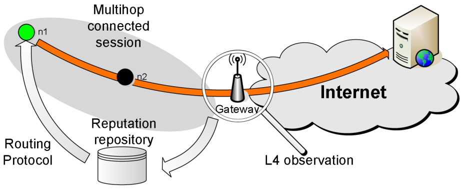


Fig. 1. Overall MOOR scheme. The service provider, or the gateway, monitors streams and reports the observations for all nodes on the route in a database that is used to compute route metrics.

After a certain period, MOOR has discovered enough ‘good nodes’ to provide connectivity to the service provider across the network; good nodes have a higher reliability metric and are used in priority. There is no need to have monitored all the nodes to establish the connectivity between the nodes and the gateway. As soon as a minimum set of cooperative nodes required to link the source node to the gateway is known, the gateway recommends this set to the source node. MOOR avoids recommending unknown and therefore potentially uncooperative relay nodes, once some 'good nodes' are available.

3.3 MOOR Components

MOOR protocol includes 2 distinct parts. The first one is pro-active and aims at discovering the nodes' neighborhood and the services provided in the network. It is called the Router Solicitation Protocol (RSP) (Figure 2, step 1). The second part is reactive; it is used to discover and evaluate several routes to the previously discovered services. It is called the Gateway Route Discovery Protocol (GRDP) (Figure 2, steps 2-3).

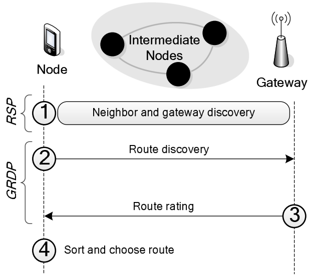


Fig. 2. MOOR protocol steps

Router Solicitation Protocol (RSP)

The Router Solicitation Protocol aims at discovering *potentially cooperative* neighbors. RSP is a protocol run locally between neighbouring nodes (i.e. with no intermediate nodes). It includes two types of control message: Router Solicitation (RS) and Router Advertisement (RA) messages[1]. A cooperative node sends RS and RA messages. We assume that a non cooperative node sends RS messages only, in order to take advantage of the network services, without providing any resource to the other nodes. A RA message contains a list of service providers associated with local metrics which indicate the minimum hop count values to reach the corresponding gateways. A node receiving an RA message stores it in an Eligible Neighbor Table until its expiry. Once RSP has converged, each node knows its closest neighbors to the gateway, in order to reach a service provider. This information is later used to discover the routes, with de help of GRDP.

MOOR Components

The GRDP provides two services: multipath discovery and end-to-end route reliability evaluation.

The route discovery process is based on Route Request (RR) messages that are forwarded by the network from the source to the destination. When a Route Request

[1] MOOR RA and RS messages shouldn't be confused with the ones related to IPV6 RFC2461.

message reaches the access gateway, the latter is in charge of computing the route reliability metric based on its local knowledge (to be explained later). The reliability metric included in the response addressed by the gateway to the source, is used by the source node to choose the most reliable discovered route.

In order to skirt non-cooperative nodes, the multipath property of the route discovery protocol is implemented as follows: an intermediate node is allowed to relay a given Route Request message, whether or not it was received from different relay nodes. As a consequence, a relay node may be part of several routes linking the same source to the same destination. This is different from the Dynamic Source Routing protocol [10] where the nodes broadcast no more than once a given Route Request message and discard the duplicates. In addition, with MOOR, each relay node forwards each received Route Request message to its N best Eligible Neighbors, previously discovered with the help of the RSP protocol. We set N equal to 2, as a compromise between overhead and performances, bearing in mind the adversarial model (no more than 1/3 malicious nodes). In addition, some appropriate forwarding rules avoid Route Request message loop formation.

By this means, a Route Request message is duplicated step by step and forwarded to its destination gateway along different routes that are no longer than 5 hops.

Node Rating and Repository

The route reliability evaluation, performed by the service provider at the gateway, is based on traffic monitoring at the transport Layer. The TCP connection control mechanism enables the detection of packet loss [8]. This technique provides different information from the one obtained with the traditional watchdog method. In particular, the gathered information cannot be imputed directly to a specific node but reveals the end-to-end behavior of the entire route. However, if we consider that the service provider is a traffic hot point – and this is actually the case in hybrid Manets – it is potentially able to qualify the nodes individually at the route intersections by carrying out crosschecks.

Our reputation system, in its current state, is implemented as follows: (i) when the first TCP packet arrives at the gateway, the system saves the route of the packet (contained in the header) and starts increasing the reputation of each relay node involved in the forwarding of that packet; (ii) if a packet is lost or on a session timeout (no FIN packet received) or if the packet route changes, the rating of each relay node involved in the incriminated route is equally decreased. This gives the following computing formula to establish the reputation Rx of a node x.

$$\begin{cases} R_x = 1 - \left[\dfrac{P_{Fail}}{P_{Fwd} + P_{Fail}} \right]^{1/k} & \text{if } P_{Fwd} > 0 \\ R_x = 0.1 & \text{if } P_{Fwd} = 0 \end{cases}$$

Formula 1 - P_{Fail} and P_{Fwd} are respectively the number of packets dropped.

The reputation value varies for each node from 0 to +1 and the gateway computes the route reliability metric by multiplying the individual node reputations. The initial node reputation value is set at 0.1.

Figure 3 represents the different components of the system and their interactions.

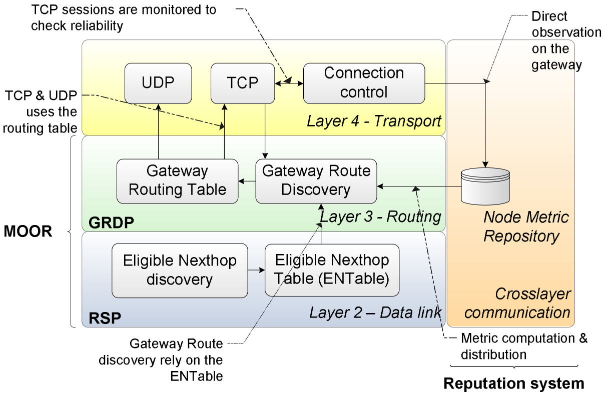

TCP sessions are monitored to check reliability
Direct observation on the gateway
UDP
TCP
Connection control
Layer 4 - Transport
TCP & UDP uses the routing table
Gateway Routing Table
Gateway Route Discovery
GRDP
Layer 3 - Routing
Node Metric Repository
MOOR
Eligible Nexthop discovery
Eligible Nexthop Table (ENTable)
RSP
Layer 2 – Data link
Crosslayer communication
Gateway Route discovery rely on the ENTable
Metric computation & distribution
Reputation system

Fig. 3. MOOR component diagram with their interactions

We are well aware of TCP protocol limitations to address link failures due to mobility and interferences when used over Manet [11]. However, we assume that any other reliable transport protocol will meet the MOOR's flow-control and error detection needs.

4 Performance Analysis

4.1 Simulation Setup

Using a reputation mechanism makes sense only if the unprotected network fails in the presence of malicious nodes. For instance, 1 or 2-hop communications – involving 0 or 1 relay nodes respectively – are still possible in non-cooperative Manets. The probability of requesting network service from a malicious node is nil for 1-hop communications and remains rather low for 2-hop communications, in presence of 1/3 malicious nodes. This is why we have decided to focus our investigations on long routes, in order to evaluate the performances of a reputation-based protocol.

To increase the hop count while keeping a rather low number of nodes, we consider a rectangular topology instead of the traditional square one. The Ad hoc network simulation size is determined to obtain a high connectivity probability in the network and to allow communications up to 5 hops. In addition, we ensure that the Ad hoc nodes are not in direct reach of the gateway. They must be at least 2 hops away, more frequently 3 hops away or even 4 or 5. This is done with the help of some fixed relay nodes located in a zone isolating the Ad hoc Network from the gateway. The malicious node percentage in the relay zone is set equal to the one in the Manet (1/3). This topology is illustrated in Figure 4.

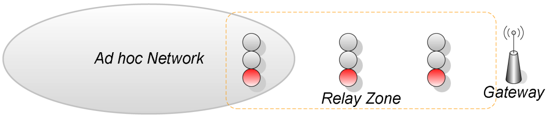


Fig. 4. A network topology to increase the average hop count to reach the gateway

A second major difference of our simulation setup lies in the use of TCP connections instead of the traditional UDP/CBR connections. We demonstrate above – in agreement with [12] - the relevance of this choice of traffic type. It is worth noting that, even if some TCP (connected) traffic is mandatory for MOOR in order to learn the network cooperation, we could have evaluated the MOOR performances on some extra UDP/CBR connections. Indeed, the cooperation knowledge acquired by MOOR with the help of TCP traffic is perfectly usable to route the UDP traffic, reliably.

The traffic scenario simulates short FTP traffic between the nodes and the gateway. The duration value of each TCP connection is 10s. The nodes initiate a communication with the gateway sequentially, one at a time. The TCP connections start at t=50s to evaluate the differences between the sleeping state and the active state. This choice of traffic scenario results in: (i) a significant number of route discoveries; (ii) a large number of nodes involved; (iii) the gateway is required to evaluate a large number of routes.

We justify our choice of sequential TCP connections, one connection at a time, in the following manner. Since by default, TCP uses all available bandwidth to transport packets, 2 simultaneous communications would share the media and if one connection cannot be established – in case of relay node misbehaving for instance – the second communication could hide this break by exploiting the unused bandwidth. Also, sequential communications allow us to avoid the already mentioned bandwidth degradation issue which is outside the scope of our study, for a better evaluation of the sole non-cooperation issue.

The other simulation parameters are the usual ones and are specified in Table 1.

Table 1. Simulation parameters

Parameters			**Value**
Nodes			30, including 21 in the Ad hoc Network and 9 in the Relay Zone
●	Normal		20, including 14 in the Ad hoc Network and 6 in the Relay Zone
●	Malicious		10 nodes (33%), including 7 in the Ad hoc Network and 3 in the Relay Zone
	●	Behavior	100% data packet dropping
Gateway			1
Connections			1 x TCP (FTP application)
	●	TCP Packet size	1020
	●	TCP window size	1
Movement			Random waypoint model
	●	Pause time	0 s (dynamic network)
	●	Speed	uniform distribution [0,10] m/s
MAC Protocol			802.11
Radio			
●	Node Tx range		250 m
●	Bandwidth		2 Mb/s
Topology			400*600 m for the Ad hoc Network
Results average (if any)			50 simulations
Simulation time			900 s

4.2 Simulation Results

Drop analysis

Figure 5 represents the packet dropping during the simulation and its impact on the data traffic in a network denoted as m7/a1. An m7/a1 network is an Ad hoc network which includes 7 malicious nodes and is protected by a reputation system running the 'a1' algorithm (an algorithm implementing reputation computation according to Formula 1). The drops are either malicious or due to mobility or collisions. We can see 3 distinctive parts in Figure 5. The first part goes from 0 to 50s for which there is no data drop since the TCP connection starts at 50s. The second part, from 50s to 250s in this particular simulation, represents the interval during which MOOR learns how well (or not) the nodes cooperate. During this period, the reputation system gathers information to compute the node reputations; the reputations are not yet accurate enough and few drops occur in the network. The last part is from 250s until the end of the simulation, during which the reputation system works well and provides good route recommendations. It results in very few malicious drops and, consequently, low impact on the throughput.

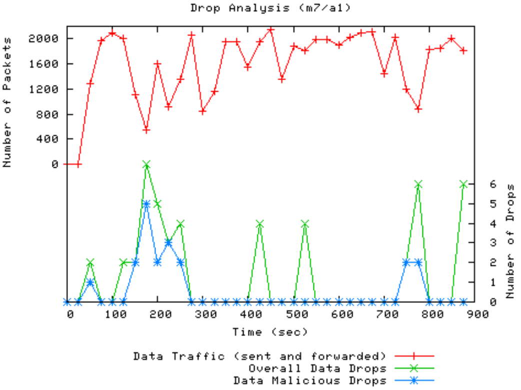


Fig. 5. Drops and the resulting throughput, in a protected network during one simulation. The malicious drops occur essentially at the beginning of the simulation which corresponds to the system initialisation time. We identify the steady state from 250s.

The same kind of results is illustrated in Figure 6, except that they relate to the corresponding network m7/a0. A m7/a0 Ad hoc network includes, in the same manner, 7 malicious nodes but is not protected by any reputation system. The 'a0' algorithm disables the reputation, by maintaining each node reputation Rx at its initial value (0.1) no matter how the nodes cooperate. As a result, nodes choose the shortest path since the route metric computed by the gateway is the multiplication of the individual node metrics (0.1n decreases with n).

Figure 6 shows that the average number of drops remains unchanged during the whole simulation and that 5 drops per timeslot make TCP communication almost impossible. The number of drops cannot be compared to the one obtained with UDP simulations since with TCP, a single drop has an important impact on the bandwidth.

As opposed to UDP, TCP is a reliable transport protocol that uses Acknowledgement packets to control the stream. If a data packet or its ACK is lost, the source stops sending packets and enters in a recovery state. After a timeout, the packet is considered lost and the source tries to send the packet again and restarts communication slowly to avoid 'congestion' on the link. As explained in [11] in Manets, packet drops are considered as congestion although this is not actually the case when we consider cooperative networks and the selfness or misbehaving issues. Therefore our investigation makes sense: without any security enforcement mechanism, nodes have major difficulties in communicating.

In addition, the comparison of Figure 5 and Figure 6 shows the contribution of a reputation system in terms of bandwidth saving.

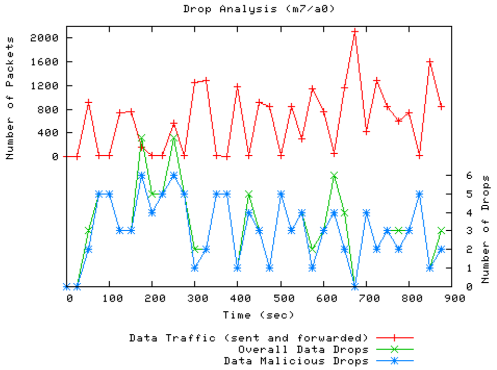


Fig. 6. Drops and resulting throughput in an unprotected network. Malicious drops occur during the whole simulation and the throughput is lower than in a protected network.

MOOR efficiency

In this part, we propose an application-motivated performance analysis. Our motivation is to evaluate if our reputation system enables or not a service provider to face the non cooperation issue when offering internet access services to remote nodes even in the presence of unavailable nodes.

In Figure 7, we analyzed the efficiency of MOOR by comparing numbers of received data packets at the destination - for the various lengths of the route followed by the sent packets (expressed as a number of hops) - both for a protected network m7/a1 and for an unprotected one m7/a0. In addition, in order to compare performances when varying the number of hops, we normalized each of these numbers of packets with its corresponding nominal number obtained with the corresponding fully cooperative network m0/a0 (no malicious nodes, no reputation system). This way, we were able to make direct comparison of the different numbers of packets obtained with a1 and a0, when varying the hop count, no matter how the network topology influences the length of the route (in our case, the most frequent hop count was 3). In addition, by evaluating the protocol performances in relative to m0/a0, we avoided underestimating the routing protocol performances in cases where no available routes existed between a pair of nodes due to network partitioning.

Figure 7 illustrates the impact of non cooperation on the number of received data packets relative to the corresponding number of data packets which would be received in the corresponding fully cooperative network, varying the distance between the source node and the gateway. Results are shown both for a protected network m7/a1 and for an unprotected network m7/a0. The confidence level on the plotted intervals is 95 %. When the route length increases, the presence of non cooperative nodes becomes a major issue in the defenseless network m7/a0. The normalized number of received data packets drastically decreases to around 40% for a 3-hop route length, and even to less than 30% for 4 and 5 hops. As soon as we introduce the MOOR reputation system, we maintain a normalized number of received data packets better than 80% for nodes distant up to 5 hops from the gateway. The high MOOR efficiency indicates that our approach enables malicious node detection and avoidance and avoids TCP stop sending packets.

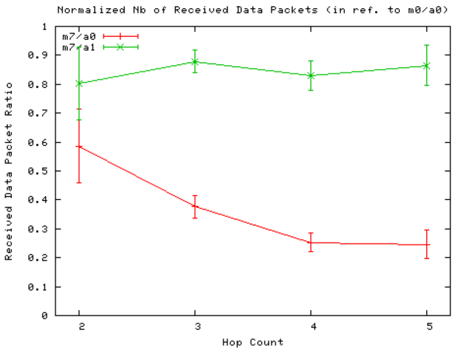


Fig. 7. Impact of hop count on the efficiency. Representation of the normalized number of data packets received at the destination for unprotected (a0) and protected (a1) networks (average of 50 simulations).

5 Extrapolation of Results and Future Work

Traditional reputation system performances may be affected in the presence of nodes dropping less than 100% packets. We showed in [13] that these latter nodes, whose reputation is not so degraded, may not be detected and considered as malicious by the usual reputation systems. This is because their reputation is evaluated individually in relation to a threshold. We expect that our end-to-end reputation system which evaluates routes relatively, should not be affected in this manner by partial misbehaving.

In the future, we would like also to investigate how reliable is our metric computation, when taking into account varying cooperative behaviors with time, instead of constant behavior. This investigation should lead to a more comprehensive reputation computation formula, for our reputation system that currently works essentially with constant node' behavior.

Conclusion

In this paper, we consider hybrid Ad hoc networks with a service provider point of view that somewhat modifies the theoretical scope and its related issues. We present a routing protocol called MOOR that makes it possible for a service provider to maximize the reliability of communication between the nodes and the operator gateway.

With the MOOR protocol, malicious nodes are avoided and skirted effectively, since it detects cooperative nodes, discovers several possible routes and qualifies them by the use of end-to-end route metrics returned to the source.

To conclude, our paper demonstrates the opportunity for an operator to maintain good connectivity, with the help of a reputation system, with nodes up to 5 hops from the gateway, in presence of 1/3 malicious nodes. In such an environment, this is not feasible if nodes relay their traffic along unqualified routes, without evaluating the cooperation of the intermediary nodes.

References

[1] Buttyan, L., Hubaux, J.: Nuglets: A virtual currency to stimulate cooperation in self-organized ad hoc networks. Technical Report, EPFL (2001)

[2] Ingelrest, F., Simplot-Ryl, D., Stojmenovi'c, I.: Routing and broadcasting in hybrid ad hoc multi-hop cellular and wireless Internet networks. Technical Report INRIA RT-291 (February 2004)

[3] Marti, S., Giuli, T., Lai, K., Baker, M.: Mitigating Routing Misbehavior in Mobile Ad Hoc Networks. In: Sixth annual ACM/IEEE International Conference on Mobile Computing and Networking, pp. 255–265 (2000)

[4] Laniepce, S., Demerjian, J., Mokhtari, A.: Cooperation Monitoring Issues in Ad Hoc Networks. In: IWCMC 2006. Proceedings of International Wireless Communications and Mobile Computing Conference, Vancouver (2006)

[5] Buchegger, S., Le Boudec, J.Y.: Performance analysis of the CONFIDANT protocol. In: MobiHoc. Proceedings of ACM International Symposium on Mobile Ad Hoc Networking and Computing, Lausanne (June 2002)

[6] Michiardi, P., Molva, R.: CORE: A COllaborative Reputation Mechanism to enforce node cooperation, Mobile Ad Hoc Networks. In: CMS 2002. Proceedings of the sixth IFIP conference on security communications and multimedia (September 2002)

[7] He, Q., Wu, D., Khosla, P.: SORI: A secure and objective reputation-based incentive scheme for ad hoc networks. In: WCNC 2004. IEEE Wireless Communications and Networking Conference, Atlanta, GA, USA (March 2004)

[8] Jensen, C.D., Connell, P.O: Trust-Based Route Selection in Dynamic Source Routing. In: Stølen, K., Winsborough, W.H., Martinelli, F., Massacci, F. (eds.) iTrust 2006. LNCS, vol. 3986, pp. 150–163. Springer, Heidelberg (2006)

[9] Li, J., Blake, C., Couto, D.S.J.D., Lee, H.I., Morris, R.: Capacity of ad hoc wireless networks. In: ACM MOBICOM (July 2001)

[10] Johnson, D.B., Maltz, D.A., Hu, Y.: The Dynamic Source Routing Protocol for Mobile Ad hoc Networks, <draft-ietf-manet-dsr-10.txt>, (expired July 19, 2004)

[11] Dyer, T.D., Boppana, R.V.: A Comparison of TCP Performance over Three Routing Protocols for Mobile Ad hoc Networks. In: ACM Symposium on Mobile Ad hoc Networking and Computing, MobiHoc (October 2001)

[12] Rajagopalan, S., Shen, C.: What does using TCP as an Evaluation Tool Reveal about MANET Routing Protocols? In: IWCMC 2006. Proceedings of the International Wireless Communications and Mobile Computing Conference 2006, Vancouver, British Columbia, Canada (July 2006)

[13] Laniepce, S., Karpagavinayagam, B., Tourolle, B.: Impacts of "On/Off" Forwarding Misbehavior on Reputation Systems in ad hoc networks. In: IWCMC 2006. Proceedings of the International Wireless Communications and Mobile Computing Conference 2006, Vancouver, British Columbia, Canada (July 2006)

Reducing the Message Overhead of AODV by Using Link Availability Prediction[*]

Liu Chao and Hu Aiqun

School of Information Science and Engineering, Southeast University, Nanjing, Jiangsu Provice, China, 210096
chao.seu@126.com, aqhu@seu.edu.cn

Abstract. To achieve high protocol efficiency is one of the primary goals in designing routing protocols for mobile ad hoc networks (MANETs), because high protocol efficiency implies low additional control overhead and power cost, which are two key issues in MANETs due to the inherited bandwidth and power-constrained characteristic. In this paper, an AODV improvement protocol with the Hello message mechanism is presented based on wireless link availability prediction. The simulation results demonstrate that the improved AODV improves the performance in terms of latency and protocol efficiency greatly, compared to the standard AODV with the periodic Hello message broadcast mechanism.

1 Introduction

In MANETs without centralized infrastructures, mobile nodes communicate with each other over multi-hop wireless links if they are out of wireless transmission range. It means that each node should commit itself to forward data packets from a neighboring node to another until a final destination is reached. Although this new approach of networking brings a great flexibility to the world of wireless communications, it imposes some strong requirements with regard to the routing functionality, because of the wireless connectivity nature of MANETs, such as high degree of mobility, absence of centralized administration and limited resources.

In [1], Macker gave some basic rules in designing Ad-hoc routing protocols, which are listed as follows:

- The protocol should be simple and efficient, since it must quickly converge to select a route before network changes make the route invalid.
- The protocol must be distributed since no centralized host is available in the network.
- The protocol should take advantage of the technology available and utilize information about the current state of the network.
- The protocol should have low control message overhead in order to preserve limited bandwidth and power resources, by minimizing route setup and maintenance messages.

[*] This work is supported by National Hi-Tech Research and Development Program of China（863 Program）No. 2006AA01Z268.

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 113–122, 2007.

As one of the most popular routing protocols for MANETs, ad hoc on-demand distance vector (AODV) has been extensively studied since it's firstly proposed by C. Perkins in [2]. To our knowledge, however, few works which address how to reduce the control message overhead in AODV have been published. In this paper, a novel method is proposed to reduce the amount of control overhead by dynamically updating the environment parameter of *Hello Interval*. The remainder of this paper is organized as follows. In Section 2, a short introduction to AODV is presented. In Section 3, we suggest an improvement method at the base of pointing out the disadvantages of periodic Hello message broadcast mechanism used in AODV. Section 4 describes the simulation model adopted, and then a detailed simulation is performed to evaluate the performance of the improved AODV. Conclusions and future work are presented in Section 5.

2 Brief Introduction to AODV

AODV is an on-demand dynamic routing protocol whose route enquiries are initiated on an on-demand basis. When a source node wishes to send a packet to a destination node for which it has no routing information in its table, it initiates a route discovery process to locate its interlocutor node by broadcasting a route request (RREQ) packet. On receiving a RREQ packet, each node first creates or updates a reverse route for itself back to the source node, and if it is neither the destination nor it has a *fresh enough* route to the destination, it rebroadcasts the RREQ packet. Otherwise, a route reply packet (RREP) is generated and unicasted back to the source node along the reverse route. Similarly, the forward route to the destination is also updated on receiving a RREP packet. When the RREP finally reaches the source node, the discovery period is terminated and the newly established route can now be used to send the data packets waiting on the buffer.

When a route toward a particular destination is created, each node along the particular path should use some available link or network layer mechanism to perform route maintenance. That is, the current node tries to know that the next hop toward the destination remains available or not. In AODV, route maintenance is usually performed by requiring nodes to send periodic Hello update messages, usually one per second. Failure of a node to receive two consecutive Hello messages assumes that the link to its neighbor is down. Subsequently, a route error (RERR) message is initiated to notify the earlier nodes down the path of such a breakage. As the RERR travels down the forward path, each affected node updates its routing table by invalidating the corresponding routes. Finally, all broken entries in the routing table of the nodes will be erased.

3 Improvement to the Standard AODV

In AODV, instead of building routes for all possible destinations in the network, a node only creates and maintains routes that it really needs. Although this on-demand approach minimizes routing table information and greatly reduces the amount of unnecessary route management overhead, it potentially leads to a large number of route requests being generated, and as a result, the increase of traffic exchange delay. To cope with this, AODV introduces a local connectivity management mechanism, namely periodic Hello message broadcast to collect information of topology changes in a timely fashion. This mechanism reduces the traffic exchange delay at the cost of increasing link connectivity management overhead which helps protocol to react quickly to changes of routes. However, periodic broadcasts of such messages have negative affects on network congestion and battery lifetime because it inevitably increases routing load of the network.

How to reduce the amount of Hello message is an interesting issue. In [3] an AODV improvement protocol with the Hello message mechanism is presented in which the function of the Hello message is expanded to the all transmission procedure of data service and management information. In other words, any transmission procedure of packets including RREQ, RREP, RERR and data packets, can be regarded as an advertisement of the presence of the transmitting node. Intuitively, this improvement can greatly reduce the amount of Hello message in scenarios with heavy traffic load, because in this case a great amount of data packets act as the role of Hello message and a Hello message is needed only when the interval period between two packets is longer than the preconfigured *Hello Interval*. This mechanism somewhat improves the performance of the protocol in terms of efficiency and latency. However, in scenarios with light traffic load, there are still a large number of Hello messages generated to perform connectivity management. So, we propose a novel method to further reduce Hello message, in which the dynamic characteristic of the network is taken into account and, instead of using periodic Hello broadcast, the use of Hello messages is adjusted dynamically according to the current condition of the network.

3.1 Concept of Dynamically Adjusting *Hello Interval*

As described above, AODV introduces a periodic Hello message broadcast mechanism to perform local connectivity management. In fact, the essence of this mechanism is to maintain a local connectivity map which will be periodically updated to monitor the actual state of wireless links between a node and all its neighbors. Fig.1 shows two typical local connectivity maps of node I. It’s obvious that in the left map, neighboring nodes far away from I are more likely to move out of the radio transmission range of it, while in the right one, because those neighbors are very close to I, the corresponding links may remain available for relative long periods. So it can be concluded that the right local connectivity map is stabler than the left one. Intuitively, when the local connectivity map of node I is in a relatively stable state, Hello messages should be broadcasted with a relatively long interval. Otherwise, node I should broadcast Hello messages with a relatively short interval so as to react rapidly to possible link breakages. If we introduce the concept of dynamically adjusting *Hello Interval* according to the current state of the network to each node, the amount of Hello messages is expected to be further reduced because unnecessary Hello messages are avoided when the local connectivity map of the considered node is in a stable state.

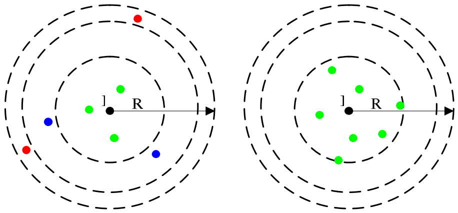


Fig. 1. Compared with green nodes (closest to I), red nodes (farthest to I) are more likely to move out of the radio transmission range of I, so the right local connectivity maps is said to be more stabler than the left one

3.2 Prediction of Lifetime Period of Wireless Links

In order to implement this concept in AODV, a link availability prediction model should be used to predict the lifetime period of wireless links based on which *Hello Interval* can be dynamically updated.

In recent years, with the rapid development of Global Positioning System (GPS) technology, it's applicable for each mobile node to be equipped with a compact, cheap and low-power consumption GPS receiver. And recently, designing routing protocols for MANETs with the aid of GPS has become a hot research topic. Similarly, we also can use the position information provided by GPS to predict the lifetime period of wireless links, which will be discussed as follows.

For simplicity of analysis, here we assume a line-of-sight (LOS) propagation model in which the path loss of signal transmission is determined by:

$$Ls = L_O + (\gamma_o + \gamma_w)d \ ,$$

$$L_O = 32.44 + 20\lg d + 20\lg f \qquad (1)$$

where L_O is the path loss in free-space propagation [4], d is the transmission range, f is the central wavelength of the signal, γ_o and γ_w are the absorption coefficients of oxygen and water vapor, respectively. Once a packet is received, the distance between the sender and the receiver, and the path loss can be easily calculated according to the location information included in the packet and received radio signal strength of it. Then we can determine the value of $\gamma_o + \gamma_w$ according to Eq. (1). Noted that since γ_o and γ_w are usually regarded as constant values in a given LOS environment, we only need to calculate $\gamma_o + \gamma_w$ for one time. Then for given transmission power P_{tx} and receiving power threshold P_{rx} , the effective radio transmission range is given by:

$$20\lg d + (\gamma_o + \gamma_w)d = P_{tx} - P_{rx} - 20\lg f - 32.44 \qquad (2)$$

From Eq.(2), we find that the path loss solely depends on the distance to the transmitter in a LOS environment. If the distance between two nodes is no more than the effective radio transmission range, we can say the corresponding wireless link is available.

Now we'll consider a simple scenario with two moving nodes, A and B, as illustrated in Fig 2. With the help of GPS, both the two nodes can acquire their position information of location, moving speed and direction periodically. Supposing B receives a packet including the position information of A, then the lifetime period of the link between A and B is given by [6]:

$$T_L^{'}(B,A)=\frac{-(ab+cd)+\sqrt{(a^2+c^2)R^2-(ad-bc)^2}}{a^2+b^2} \tag{3}$$

where $a=v_i\cos\theta_i-v_j\cos\theta_j$, $b=x_i-x_j$, $c=v_i\sin\theta_i-v_j\sin\theta_j$, $d=y_i-y_j$, and R is the effective radio transmission range determined by Eq.(2).

3.3 Link Availability Prediction Based AODV

For a considered node, if the *Hello Interval* is selected according to formula (3), it is sure that the corresponding wireless link between the two nodes will not break due to node movements. Nevertheless, the interval between two Hello messages should not be very long, otherwise the protocol can not response quickly in the case of node power off or its moving into the coverage of unknown nodes. So the lifetime period of the wireless link between two nodes can be selected according to the subsection function:

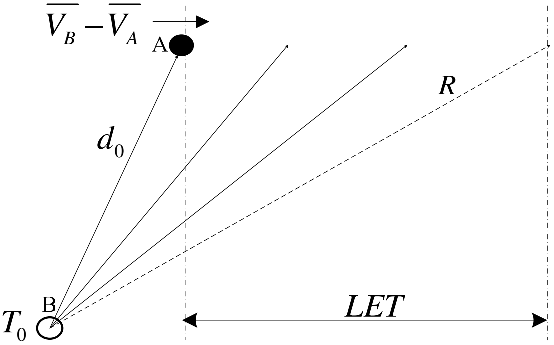


Fig. 2. Link lifetime period prediction of two nodes

$$T_L(i,j)=\begin{cases}1 & T_L^{'}(i,j)<1\\ T_L^{'}(i,j) & 1\le T_L^{'}(i,j)\le T_{threshold}\\ T_{threshold} & T_L^{'}(i,j)>T_{threshold}\end{cases} \tag{4}$$

Intuitively, the period of Hello message broadcast is mainly determined by the link with the shortest lifetime period. So, we propose a simple model to determine the *Hello Interval* for a considered node i :

$$T_h(i) = \min\{T_L(i, j)\}\,,$$

$$j \in \{neighbor\ set\ of\ node\ i\} \tag{5}$$

where $T_L(i, j)$ is the lifetime period of link between node i and its neighbor j given by Eq. (4).

In order to evaluate the simulation performance of AODV with the improved Hello message mechanism described above, some modifications must be added to the AODV simulation model, which can be illustrated in Fig.3.

```
Initialization:
    {A self-interrupt event is scheduled in $T_h(i)$ = $T_{threshold}$ seconds of the current
    simulation time to initiate the first Hello broadcast.}
Wait until an event interrupt occurs:
IF (INTRPT_TYPE == HELLO_INTERRUPT)
    {Another self-interrupt event is scheduled in $T_h(i)$ = $T_{threshold}$ seconds of the
    current simulation time for the next Hello broadcast.}
ELSEIF (INTRPT_TYPE == PK_ARVL_FROM_MAC)
        {Get the position information from the received packet;
    Calculate the lifetime period between the current node and its neighbor from
    which this packet arrived according to Eq.(3);
    /* Update the Hello Interval if $T_h(i)$ is less than the remainder of time-out period
    of the current Hello Interval timer. */
    IF ($T_h(i)$ < remain_hello_time-out_period)
        Invalidate the current hello interrupt event and reschedule a new one in $T_h(i)$
        seconds of the current simulation time.
        }
ELSEIF (INTRPT_TYPE== PK_ARVL_FROM_APPL)
        {Transmit the packet to the MAC layer.
         Invalidate the current hello interrupt event and reschedule a new one in
    $T_h(i)$ = $T_{threshold}$ seconds of the current simulation time.}
ELSE return;
```

Fig. 3. Modifications to AODV simulation model

4 Simulation Model and Results

In this section, a series of simulation experiments in OPNET [5] network simulator will be conducted to perform an evaluation analysis on the performance ability of AODV with the three discussed mechanisms, namely periodic Hello message based (SAODV), the function expanded Hello message based (EAODV), and the link availability prediction based (PAODV), which will be introduced in details as follows.

4.1 The Simulated Network Scenario and Model

All simulations were conducted for 50 seconds of simulation time in a network scenario with 30 nodes uniformly distributed in an area of 1500m x 1500m. Each node moves within the area, with a random direction and a random velocity uniformly distributed between 0 and a maximum value of 10m/s. And for simplicity, the parameter $T_{threshold}$ is set to 4 seconds in our simulations.

As the IEEE 802.11 standard is widely used in test beds and simulation platforms, we use The Distributed Coordination Function (DCF) of IEEE 802.11 as the MAC layer protocol in our simulations.

The PHY model is modeled as a shared-media radio with a bit rate of 1 Mb/s and a radio transmission range of 300m.

Finally, the background traffic is modeled by 30 source nodes generating packets with inter-arrival time uniformly distributed between 0 and a maximum value of 8.0 seconds, and the packet size is 256 bytes.

4.2 The Performance Metrics

The follow two important performance metrics are chosen to assess the improved AODV protocol:

- Average End-to-End Delay(Latency) — This includes all possible delays caused by buffering during route acquisition time, queuing at the interface queues, retransmitting at the MAC, and propagation and transmission delay, etc.
- Normalized Routing Protocol Efficiency — The amount of data traffic generated (in bits) per total traffic delivered (in bits), including data traffic and control traffic messages (RREQ, RREP, RERR and Hello). Each hop-wise transmission of a routing control message is counted as one transmission.

4.3 Simulation Results and Technical Analysis

This subsection presents the simulation results and analysis of the relative performance of AODV with the three mechanisms discussed above.

Fig.4 shows the relative latency performance of the three different versions of AODV. From this figure, we can see that PAODV gives the best latency performance, followed by EAODV and SAODV. It’s because PAODV further reduces routing overhead and performs the local connectivity management with a more efficient way, compared to its counterparts.

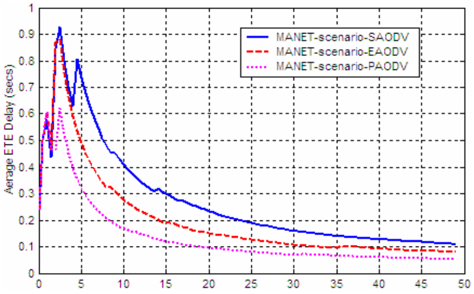


Fig. 4. Variation in Average End-to-End Delay

Fig.5 shows the variation of normalized routing protocol efficiency of AODV with the three different mechanisms, respectively. We observe that, PAODV has the best protocol efficiency performance, while SAODV has the worst, followed by EAODV. It can be explained by the fact that periodic Hello message broadcast in SAODV considerably increases the routing load in the network, and thus severely degrades the protocol efficiency. EAODV expands the function of the Hello message to the all transmission procedure of data service and management information, so the routing load is greatly decreased. In PAODV, a link availability prediction algorithm is added to AODV to predict the lifetime period of a wireless link based on which Hello period can be dynamically selected, so the routing load is further decreased, thus resulting higher protocol efficiency.

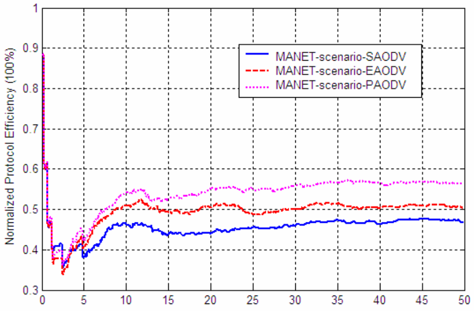


Fig. 5. Variation in Normalized Protocol Efficiency

Fig.6 shows the variation of protocol efficiency performance of AODV with the three different mechanism when the average packet inter-time ranges from 0.25 seconds to 4.0 seconds. We find that when the packet inter-time is big, which means low data packet generating rate, EAODV gives slightly better efficiency performance than SAODV. It's because in this case, only a few data packets are generated to act as the role of Hello messages. While when the packet inter-time is small, which means high data packet generating rate, EAODV and PAODV gives almost similar

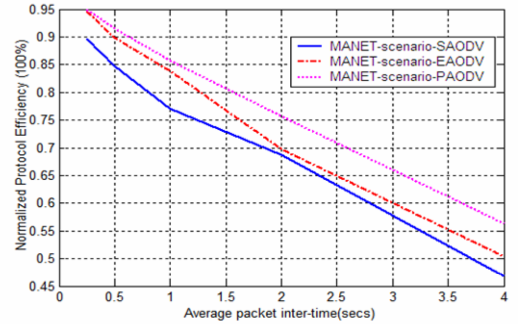


Fig. 6. Normalized Protocol Efficiency for varying average packet inter-time

performance in terms of protocol efficiency. It's because frequent data packets transmission procedures lead to a large number of Hello messages being reduced, and the link availability prediction mechanism almost can not further reduce the Hello broadcast messages.

5 Conclusions and Future Work

In this paper, we have analyzed the standard AODV and pointed out the disadvantages of periodic Hello message broadcast mechanism which is usually used in AODV for local connectivity management. According to the analysis, we presented an improved method that *Hello interval* dynamically updates based on the wireless link availability. Simulation results show that the improved AODV further decreases the routing overhead, thus resulting better performance in terms of latency and routing protocol efficiency.

It's noted that the proposed wireless link availability prediction algorithm is considered in a LOS propagation environment. While in a typical wireless environment where the transmitted signal over the wireless channel is subject to multi-path fading, the prediction may not be accurate. However, our work figure out a novel way to improve the efficiency of routing protocols for MANETs. How to provide a simple, efficient and economic way to make prediction in the context of typical wireless fading channel is still a challenge, which should be deeply studied in future.

References

1. Macker, P., Corson, M S.: Mobile Ad hoc Networking and the IETF. Mobile Computing and Communications Review 3(1), 11–13 (1999)
2. Perkins, C E., Royer, E M.: Ad Hoc on demand distance vector routing[A]. In: Proceedings of 2nd IEEE Workshop on Mobile Computing Systems and Applications [C], pp. 90–100 (1999)

3. Xiaoshu, C., Xia, L.: Improved high efficiency AODV routing protocol for Ad Hoc networks. Journal of Southeast University (Natural Science Edition) 33(2) (March 2003)
4. Rappaport, T.S.: Wireless Communications: Principles and Practice. Prentice Hall, Upper Saddle River, NJ (1995)
5. OPNET. MIL 3 Inc. [Online]. (A free version of OPNET Modeler is offered by OPNET College Program for research and education purpose), Available: http://www.opnet.com
6. Lee, S.J., Su, W., Geral, M.: Ad hoc wireless networks with mobility prediction [C]. In: Proc IEEE ICCCN 1999, Boston, MA, pp. 4–9 (1999)

An Energy-Efficient and Low-Latency Sink Positioning Approach for Wireless Sensor Networks

Fanrui Kong[1], Chunwen Li[1], Xuedong Zhao[2], Qingqing Ding[2],
Fei Jiao[2], and Qibin Gu[2]

[1] Department of Automation Tsinghua University, Beijing 100084, China
[2] Department of Electrical Engineering Tsinghua University, Beijing 100084, China
{kongfr,d-qq00,jiaof05,gqb03}@mails.tsinghua.edu.cn
{lcw,eeazhaoxd}@mail.tsinghua.edu.cn

Abstract. This paper investigates the sink positioning problem in Wireless Sensor Networks(WSNs) with the consideration of the energy-latency trade-offs. Energy-efficiency and low-latency are two major objectives in most researches on WSNs. Positioning the sink properly and exploiting its mobility can improve the two performances. A novel linear programming model is proposed to solve the sink positioning problem. Its objective function represents the overall performance of the network lifespan and the average packet latency. We can get not only the position pattern of the sink but also the sojourn time ratio for each possible position according to the optimization results. Simulations are accomplished on NS-2. The results show that compared with a static sink approach or a positioning approach which only concerns the energy-efficiency, our approach can greatly shorten the average packet latency and prolong the network lifespan, especially when the sensor nodes are distributed asymmetrically or the traffic load is unbalanced.

Keywords: wireless sensor networks, sink position, low-latency, energy-efficient, lifespan.

1 Introduction

Wireless sensor network is a self-organized network consisting of hundreds or thousands of sensor nodes. A sensor node is usually equipped with a sensing device, a low capacity processor, a short-range wireless transceiver and a limited energy supply[1, 2, 3]. These sensor nodes communicate in an ad hoc manner and collaborate to achieve a certain task. The development of low-power and low-cost electric devices makes it possible for WSNs to accomplish various applications such as event detecting, environment monitoring, intelligent building, space exploration, disaster relief and some other areas[4, 5, 6].

The sink is the data collector of WSNs. Sensor nodes obtain wanted data from the environment and send it to the sink. At the sink, data is integrated, compressed, and transmitted to the users by infrastructures like LAN, Internet

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 123–134, 2007.

or satellite. Compared with the sensor node, the sink has a relatively infinite energy-supply, stronger communicating and processing ability.

Due to the energy constraints of the sensor nodes, most researches on WSNs focus on the protocols or schemes aiming at saving sensor nodes' limited energy to prolong the network lifespan[7, 8]. Nowadays as the applications of WSNs become various and the capability of the micro-electric devices improves, a new researching area focusing on the Quality of Service (QoS), especially the performance of the packet latency, is being payed more and more attention to. In some applications, it is even more crucial to shorten the packet latency than to prolong the network lifespan. For example, in mine-safety monitoring system the packet with the peril data must be send to the supervisor within a prescript time, since any unnecessary delay may result in an accident or even injury and death. Many researchers have done a great deal of works to shorten the packet latency and put forward many applicable methods[9, 10, 11, 12, 13]. Employing the mobility of the sink is an important one. Since the feature of the sink is unique that all the nodes have to send their packets there, positioning the sink properly and exploiting its mobility can facilitate the performance improvement of WSNs.

In this paper, we present a sink positioning approach to achieve the optimization of the synthetic performance of both packet latency and network lifespan. The position of the sink is strongly relevant to the average number of the hops from a sensor node to the sink, which is a crucial factor in shortening the packet latency when the network topology and the traffic load are fixed. The position of the sink also greatly affects the network lifespan. Since sensor nodes relay their packets to the sink, the nodes near the sink usually have more packets to transmit, consume more energy and die faster than remote nodes. So balancing the energy consumption of all the nodes can improve the performance of network lifespan. Sink mobility can help to achieve such balancing. Our approach investigates the sink position pattern and sojourn time ratio to achieve both energy-efficiency and low-latency.

The remainder of this paper is organized as follows: In section 2 some related works are introduced and the difference between them and our approach is proposed. System model is outlined in section 3, including the network assumptions, the energy consumption model and the latency model. Problem is formulated in section 4 in the form of linear programming. Section 5 consists of the main simulation results and the performance evaluation of our approach. Conclusions are drawn in section 6 and some opening issues are outlined.

2 Related Works

In [14] the authors propose an algorithm to get the optimal placement of the sink maximizing network lifespan. It requires $M = O((n/\varepsilon^3)^2 log^2(n/\varepsilon))$ preprocessing and needs to solve M instances of linear programming, where n is the number of the nodes and ε is a predetermined positive number. It also gives a faster algorithm. This paper presents that the optimal position of the sink can be solved by linear programming.

Jianping Pan [15] considers generic two-tiered WSNs consisting of sensor clusters, including two kinds of nodes, Sensor Nodes(SNs) and Application nodes (ANs). The sink is relatively flexible. SNs capture, encode, and transmit relevant information from a designated area within a sensor cluster, and ANs receive raw data from these SNs, create a comprehensive local-view, and forward the composite bit-stream toward the sink. The optimization algorithm is based on the assumption that all the ANs can communicate with the sink within one hop, which is unrealistic when the sensing area is broad.

Chang and Tassiulas[16, 17] consider a more general version of the problem. They formulate a routing multi-commodity flow problem with node capacities as a linear program and their objective is to maximize the lifespan of the system. In [16], the authors note that if the transmitted power level and the node capacity at each node are fixed, then the problem is equivalent to a maximum flow problem.

The works above are intended to determine the optimal position of the sink, and then the sink is fixed there.

Shashidhar Rao Gandham[18] proposes a method of employing the mobility of the sink and presents that employing the mobility of the sink can achieve much longer lifespan than a static sink. He uses a linear programming model to determine new locations for the sink and a flow-based routing protocol to ensure energy efficient routing during each round. But the author doesn't explicitly give out how long each round should last.

Z.Maria Wang[19] proposes the approach to obtain the sojourn time at each position of the sink on the basis of [18]. Only the lifespan is concerned and optimized. The analytical results are supported by a network of a grid topology. The sink positioning approach in this paper may result in a great increase in packet latency especially in the situation where the nodes are randomly deployed and the traffic pattern is asymmetrical.

[18, 19] exploit the mobility of the sink, but they optimize the sink position only with the objective to save the nodes' energy or to prolong the network lifespan.

Our paper considers not only the energy-efficiency but also the packet latency performance. To the best of our knowledge, this is the first paper which employs a latency-concerned network lifespan objective function to optimize the sink movement pattern and the sojourn time at each position.

3 System Model

In this section a simplified system model is presented which is used to set up the linear programming model of our approach.

3.1 Network Assumptions

There are many factors that can affect the energy efficiency and packet latency. But in our approach only the sink position is concerned and investigated. So we make the following simplifying assumptions in building the system model.

- All the sensor nodes are static. There is only one sink in the network. The sink is mobile and can stay at a group of possible positions in the network.
- The initial energy of all the sensor nodes is the same and can't be recharged.
- Sensor nodes are homogeneous with a fixed radio range. The wireless channels are bi-directional, symmetric and error-free.
- Only the energy consumption for communication is considered. The energy consumption during idle time is neglected.
- The network operates under a certain Medium Access Control (MAC) protocol and a certain multi-hop routing protocol.

3.2 Energy Consumption and Network Lifespan Model

First order radio model[20] is used to formulate the energy consumption of each sensor node. In both transmitter and receiver circuitry, the energy consumed is $E_{elec} J/bit$. The energy consumed in amplifier of the transmitter is $\varepsilon_{amp} J/bit/m^2$. Thus the energy consumed for transmitting a bit of data to distance of d away, E_{TX}, and the energy for receiving a bit, E_{RX}, are respectively

$$E_{TX} = E_{elec} + \varepsilon_{amp} \times d^2 \tag{1}$$

$$E_{RX} = E_{elec} \tag{2}$$

In most applications, d is the maximum distance a packet reaches, which defines the radio range. It is assumed to be a constant value. So E_{TX} and E_{RX} are constant too.

There are many models for the network lifespan such as N-of-N lifetime, K-of-N lifetime and m-in-K-of-N lifetime[18]. In our model we use the N-of-N lifetime. The network fails when any single sensor node runs out of the energy so the network lifespan is denoted as $L\ (s)$:

$$L = min\{l_i\} \tag{3}$$

$l_i\ (s)$ is the life of node i.

3.3 Packet Latency Model

Most WSNs operate in an ad hoc way. Sensor nodes are both data-generators and data-routers. The sensor node obtains the wanted sensing data and then transmits it to a specific neighbor (node in the radio range) which is determined by the routing protocol. Obviously multi-hop communication can lead to an increase of the packet latency. The more hops are needed, the longer packet latency is.

As all the nodes share only one wireless medium, the MAC protocol should ensure that the nodes within the radio range can take turns to use the channel. When a node has some packets to transmit and the medium is busy, it has to wait to contend for the medium until the medium is idle. Of course there are chances that after the medium is idle, more than one nodes want to transmit their packets and only one node can win to access the medium. So the sensor node may have to wait for many rounds before its turn to transmit.

These two specific features of WSNs, multi-hop communication and contending strategy, are factors that significantly influence the packet latency. We define them as *hop factor* and *queuing factor*. *Queuing factor* determines the latency within one-hop communication, while *hop factor* determines how many rounds of queuing a packet has to stand with along its path from its initiator to the sink. The relationship between these two factors and the latency is much like the one between the price, the quantity and the expenditure.

Queuing factor is strongly relevant to the MAC protocol, the sensor nodes density and the sensor nodes configuration. Sensor nodes are randomly and redundantly deployed in most WSNs applications by ejection from an plane or other similar ways. So the network density and the sensor nodes configuration are usually uncontrollable. There are many works devoted to MAC protocols' effect on packet latency. But it is out of the range of this paper.

Hop factor critically depends on the routing protocol, sensor nodes density, sensor nodes configuration and the sink position. More hops usually mean more energy consumption. Most routing protocols try to achieve energy-efficiency and actually, though not deliberately, reduce the hops from the transmitting sensor node to the sink. Designing a proper routing protocol is also out of the scope of this paper. The network density and the sensor nodes configuration are hard to manipulate, but the sink is usually controllable. The sink can be located artificially and may be equipped with mobile vehicles to move freely in the sensing area. So optimizing the configuration of the sink is a good method to affect the *hop factor*.

Since the sink location pattern and the sojourn time only influence the *hop factor*, we only consider the number of the hops from the sensor node to the sink in our latency model. We define h_k^i as the number of the hops from sensor node i to the sink, which is located at position k.

4 Problem Formulation

Linear programming is used in this paper to optimize the latency-concerned lifespan. The network is modeled as a graph $G(N, M, R)$. N is the set of the sensor nodes and the number of its elements is n. M is the set of the possible sink positions and the number of its elements is m. $R \subseteq N \times N$ represents the routing scheme used in the network and its element $(i, j) \in R$ means that node i sends its packets to node j according to its routing table.

The optimization variable is t_k (s), which is the sojourn time when the sink settles at position k $(k \in M)$. h_k^i is the number of the hops from sensor node i to the sink whose position is k. h_k^i can be denoted as:

$$h_k^i = 1 + h_k^j \qquad (i, j) \in R \tag{4}$$

H_k is defined as the average number of the hops from all the sensor nodes to the sink when the sink settle at position k.

$$H_k = \frac{\sum\limits_{i \in N} h_k^i}{n} \tag{5}$$

Prolonging the network lifespan and reducing the average packet latency are pursued. So we use the metric t_k/H_k to evaluate the synthetic performance with the consideration of both energy-efficiency and low-latency. The objective function is :

$$\sum_{k \in M} \frac{t_k}{H_k} \tag{6}$$

Since the energy of each node is limited and can't be recharged, the energy consumption is the constraint of the optimization model. So the linear programming model can be depicted as following:

$$MAX \qquad \sum_{k \in M} \frac{t_k}{H_k} \tag{7}$$

constraints:

$$\sum_{k \in M} (r_k^i(t) \cdot P_{TX} + r_k^i(r) \cdot P_{RX}) \cdot t_k \leq E_0 \tag{8}$$

$$t_k \geq 0 \qquad\qquad k \in M \tag{9}$$

where E_0 (J) is the initial energy of each sensor node

P_{TX} $(J/packet)$ and P_{RX} $(J/packet)$ are the energy consumptions for transmitting and receiving a packet.

$$P_{TX} = E_{TX} \cdot Le \tag{10}$$

$$P_{RX} = E_{RX} \cdot Le \tag{11}$$

where $Le(bit/packet)$ is the length of a packet

$r_k^i(r)$ $(packet/s)$ and $r_k^i(t)$ $(packet/s)$ are the receiving and transmitting rate of sensor node i when the sink resides at position k.

$$r_k^i(r) = \sum_j r_k^j(t) \qquad\qquad (j,i) \in R \tag{12}$$

$$r_k^i(t) = r_k^i(r) + r_k^i(g) \tag{13}$$

$r_k^i(g)$ $(packet/s)$ is the data generation rate of sensor node i

5 Simulation Results and Performance Evaluation

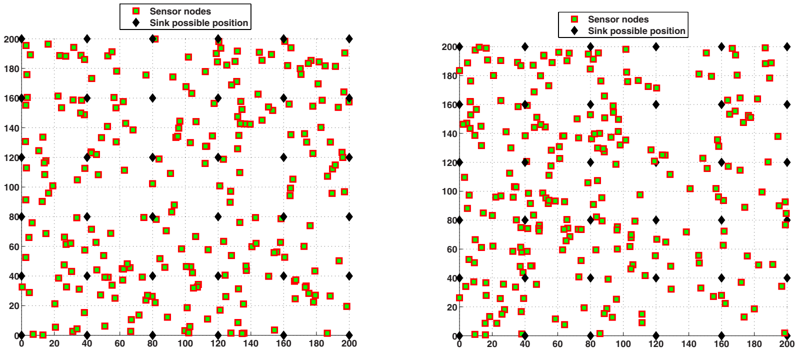
5.1 Simulation Setup

As showed in Fig.1, we built a simulator where 240 sensor nodes are scattered in an square sensing area of 200 × 200 m^2 and the possible sink positions are on the intersections of an 6 × 6 grid. The grid covers the square sensing area. IEEE 802.11[21] and directed diffusion[22] are engaged as the MAC protocol and the routing scheme. The radio range is set to be 40 m. The simulations are all done on NS-2.28. The value of the parameters are $E_0 = 10J, P_{RX} = 0.000087J/packet, P_{TX} = 0.00018J/packet$.

In order to evaluate the performance of our approach, we compare it with other two typical cases to demonstrate how our approach improves both the energy conserving performance and the latency shortening performance.

1. Sink is stationary at the center of the sensing area
2. Sink is mobile and moves according to the approach of [19]
3. Sink is mobile and moves to maximize the latency-concerned network life time

Case 3 represents our approach.



(a)sensor nodes distributed uniformly

(b)160 nodes in the left half
80 nodes in the right half

Fig. 1. Network topology with 240 nodes in an square area of 200×200 m^2

The average packet latency in case 1 usually approximates optimal value when the nodes are uniformly distributed and the traffic is balanced, since locating the sink at the center can keep the network topology more symmetric. The purpose of choosing this case is to contrast the latency performance of our approach with the approximately optimal one. In case 2, the approach of [19], only the energy-efficiency is concerned and optimized. In case 3, which represents our approach (case 3 and our approach are used interchangeably), the sink moves round by round and every single round lasts exactly an hour. A round is divided into small time slices according to the sink position and the ratio of the sojourn time obtained by solving the linear programming problem (7) to (9). And then the sink settles at the possible position for a period of its time slice. For example, if the results of the linear programming problem (7) to (9) is that the sink settles at position a, b, and c and the sojourn time are 1000, 2000 and 3000 seconds respectively, then the sojourn time ratio of the possible positions is 1 : 2 : 3 and the sink will settle at position a, b, and c for 10, 20, and 30 minutes every round.

During each round, the sink moves clockwise from the center part of the sensing area to the peripheral part.

In order to evaluate the performance of our approach under the condition of different nodes distribution and traffic load pattern, we do the simulations for the following three circumstances:

1. Uniform nodes distribution and balanced traffic load
2. Weighted nodes distribution and balanced traffic load
3. Uniform nodes distribution and unbalanced traffic load

We also accomplish some parallel simulations with more possible sink positions.

5.2 Uniform Nodes Distribution and Balanced Traffic Load

All the nodes are uniformly distributed in the sensing area and the data generation rate of each sensor node, $r_k^i(g)$, is the same. The network topology is showed in Fig.1(a). $r_k^i(g)$ is set to be 1 $packet/s$. The sink location pattern and the sojourn time ratio is showed in TABLE 1(a). It is obtained by solving the linear programming problem (7) to (9).

Fig.2(a) shows the simulation results of the lifespan and the average packet latency. Since nodes are distributed uniformly and the sink is located at the symmetric center of the sensing area, the average packet latency is the shortest in case 1. But the lifespan of case 1 is remarkably shorter than the other two cases, since a mobile sink can lead to more even consumption of the nodes' energy. Our approach also outperforms case 2 (the work of [19]) in both lifespan and average packet latency. Firstly, this is because the nodes are distributed randomly, which is more realistic than the assumption of case 2 that the network is like a grid. Secondly, the sink pattern and the sojourn time are obtained by maximizing the synthetic performance of both energy-efficiency and low-latency in our approach, but in case 2 only energy consumption is concerned.

The average packet latency of our approach is 4.07% longer than that of case 1, but 19.6% shorter than that of case 2. The lifespan of our approach is 3.32 time and 0.841 times longer than that of case 1 and case 2 respectively.

5.3 Weighted Nodes Distribution and Balanced Traffic Load

As showed in Fig.1(b), 160 sensor nodes are located randomly at the left half part of the sensing area, and the other 80 sensor nodes are located at the right half part. $r_k^i(g)$ is still set to be 1 $packet/s$ for all the sensor nodes. TABLE 1(b) shows the sink pattern and the sojourn time ratio.

From TABLE 1(b) we can find that the sink explicitly settles more often and longer in the left half where the sensor nodes are more densely deployed. Fig.2(b) shows the lifespan and average packet latency of the three cases. The average packet latency of our approach is the shortest of all three cases. The packet latency of case 1 becomes longer, because the symmetric center of the network topology no longer lies at the center of the sensing area due to the weighted nodes distribution.

The average packet latency of our approach is 13.6% shorter than that of case 1, and 22.5% shorter than that of case 2. The lifespan of our approach is 3.33 time and 1.21 times longer than that of case 1 and case 2 respectively.

5.4 Uniform Nodes Distribution and Unbalanced Traffic Load

The topology of the networks is the same as Fig. 1(a), but the traffic load is unbalanced. $r_k^i(g)$ of the sensor nodes in the left half part is set to be 1 $packet/s$, while 3 $packet/s$ for the right half part. TABLE 1(c) shows the sink pattern and the sojourn time ratio. The sink settles more often and longer in the right half part because the traffic load there is heavier than the left half. Our approach also outperforms the other two cases. From Fig. 2(c), we can see that the average packet latency of our approach is 7.6% shorter than that of case 1, and 30.4% shorter than that of case 2. The lifespan of our approach is 3.85 time and 1.51 times longer than that of case 1 and case 2.

Table 1. Sink location pattern and the sojourn time ratio

0	0	0	1.18	0	0
0	3.21	0	3.14	2.39	0
0	1.83	3.07	2.70	2.33	0
0	0	3.73	3.24	0	0
0	1	2.90	2.31	0	0
0	0	0	0	0	0

(a)uniform nodes distribution and balanced traffic load

0	0	0	0	0	0
0	1	1.15	1.11	0	0
1.25	1.49	3.11	1.13	0	0
0	1.88	4.04	1.73	1.13	0
0	1.23	1.11	1.84	0	0
0	0	0	0	0	0

(b)weighted nodes distribution and balanced traffic load

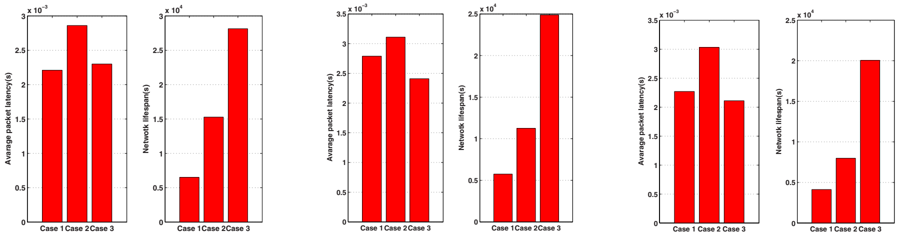
0	0	0	0	0	0
0	0	0	2.46	0	0
0	1.30	2.14	4.82	2.83	1.45
0	0	3.32	3.32	3.76	0
0	0	1.21	2.67	0	0
0	1	0	0	0	0

(c) uniform nodes distribution and unbalanced traffic load

5.5 Parallel Simulations with More Possible Sink Positions

Another four parallel simulations are done with the possible sink positions on the intersections of 8 × 8,10 × 10,12 × 12 and 14 × 14 grid. The lifespan and the packet latency of different grid scale is shown in Fig. 3 and Fig. 4. The performances of the network in case 1 doesn't vary much with the grid scale, because the sink is static. The network lifespan increases in case 2 and case 3, because more possible sink positions leads to more various position pattern and more even energy consumption. The average packet latency of case 2 does't vary much since latency is not concerned there. But the average packet latency of case 3 decrease explicitly, due to the more possible sink positions, which leads to more optimized results of the linear programming problem.

From Fig. 3 and Fig. 4, the advantage of our approach to the other two cases becomes more significant with the increase of the grid scale.

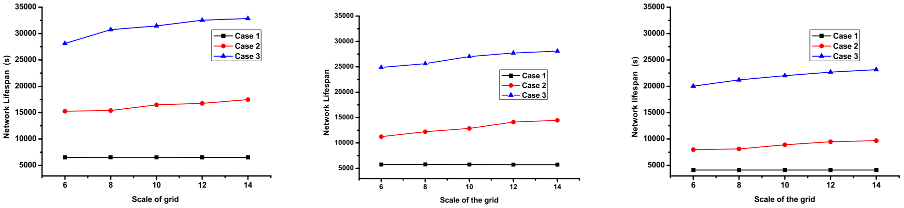


(a)uniform nodes distribution and balanced traffic load

(b)weighted nodes distribution and balanced traffic load

(c) uniform nodes distribution and unbalanced traffic load

Fig. 2. Performance of average packet latency and network lifespan

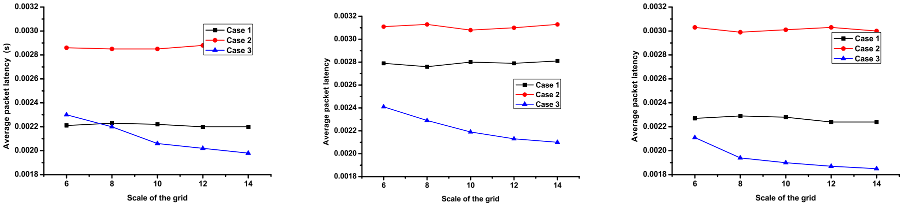


(a)uniform nodes distribution and balanced traffic load

(b)weighted nodes distribution and balanced traffic load

(c) uniform nodes distribution and unbalanced traffic load

Fig. 3. Network lifespan versus grid number



(a)uniform nodes distribution and balanced traffic load

(b)weighted nodes distribution and balanced traffic load

(c) uniform nodes distribution and unbalanced traffic load

Fig. 4. Average packet latency versus grid number

6 Conclusion and Future Work

In this paper, we have proposed a novel sink positioning approach based on linear programming, which optimizes the synthetic performance of both average packet latency and network lifespan. The sink position pattern and the sojourn time ratio for each position is obtained by solving the linear programming problem. Two typical cases are contrasted with our approach. The simulation results show that compared with these two cases, our approach can greatly shorten the packet latency and prolong the network lifespan, especially when the sensor nodes are distributed asymmetrically or the traffic load is unbalanced. Furthermore the advantage of our approach becomes more significant as the number of the possible sink position increases.

In the future works, we aim to investigate the sequence of the sink movement and the schemes to handle the changing network topology while the sink moves.

References

[1] Qia, H., Iyengarb, S., Chakrabartyk, K.: Distributed sensor networks-a review of recent research. Journal of the Franklin Institute 338, 655–668 (2001)

[2] Aboelaze, M., Aloul, F.: Current and future trends in sensor networks: A survey, Sydney. In: Second IFIP International Conference on Wireless and Optical Communications Networks, pp. 551–555 (2005)

[3] Akyildiz, I., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless sensor networks: a survey. Computer Networks 38, 393–422 (2002)

[4] Sohrabi, K., Gao, J., Ailawadhi, V., Pottie, G.J.: Protocols for self-organization of a wireless sensor network. Personal communications 7, 16–27 (2000)

[5] Chong, C.Y., Kumar, P.: Sensor networks: evolution, opportunities, and challenges. Proceedings of the IEEE 91, 1247–1265 (2003)

[6] Faza, A., Sedigh-Ali, S.: A general purpose framework for wireless sensor network applications. In: COMPSAC 2006. 30th Annual International Computer Software and Applications Conference, Chicago, USA, pp. 356–358 (September 2006)

[7] Ye, W., Heidemann, J., Estrin, D.: An energy-efficient mac protocol for wireless sensor networks. In: INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings, vol. 3, pp. 1567–1576. IEEE, Los Alamitos (2002)

[8] Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H.: Energy-efficient communication protocol for wireless microsensor networks. In: Proceedings of the 33rd Hawaii International Conference on System Sciences, Howaii, USA, vol. 2 (January 2000)

[9] Das, S., Choi, W.: Coverage-adaptive random sensor selection with latency control for application-specific data gathering in wireless sensor networks, vol. 1, pp. 214–219 (June 2005)

[10] Zhu, J., Papavassiliou, S., Kafetzoglou, S., Yang, J.: An efficient qos-constrained data aggregation and processing approach in distributed wireless sensor networks. In: ISCC 2006. Proceedings. 11th IEEE Symposium on Computers and Communications, pp. 257–262 (June 2006)

[11] Padmanabh, K., Roy, R.: Multicommodity flow based routing in wireless sensor network with lifetime latency tradeoff, pp. 1–10 (January 2006)

[12] Ruzzelli, A., Tynan, R., O'Hare, G.: An energy-efficient and low-latency routing protocol for wireless sensor networks. In: Proceedings Systems Communications, 2005 (August 2005)

[13] Ruzzelli, A., Evers, L., Dulman, S., van Hoesel, L., Havinga, P.: On the design of an energy-efficient low-latency integrated protocol for distributed mobile sensor networks. In: International Workshop on Wireless Ad-Hoc Networks, 2004, pp. 35–44 (June 2004)

[14] Efrat, A., Har-Peled, S., Mitchell, J.S.B: Approximation algorithms for two optimal location problems in sensor networks. In: 2nd International Conference on Broadband Networks, pp. 714–723 (October 2005)

[15] Pan, J., Cai, L., Hou, T., Shi, Y., Shen, S.X.: Optimal base-station locations in two-tiered wireless sensor networks. Mobile Computing 4, 458–473 (2005)

[16] Chang, J.H., Tassiulas, L.: Energy conserving routing in wireless ad-hoc networks. In: Proceedings of IEEE Infocom2000, pp. 22–31 (October 2000)

[17] Chang, J.H., Tassiulas, L.: Fast approximate algorithms for maximum lifetime routing in wireless ad-hoc networks. Mobile Computing 4, 458–473 (2005)

[18] Gandham, S.R, Dawande, M., Prakash, R., Venkatesan, S.: Energy efficient schemes for wireless sensor networks with multiple mobile base stations. In: Global Telecommunications Conference 2003, pp. 377–281 (December 2003)

[19] Wang, Z.M., Basagni, S., Melachrinoudis, E., Petrioli, C.: Exploiting sink mobility for maximizing sensor networks lifetime. In: Proceedings of the 38th Hawaii International Conference on System Sciences (January 2005)

[20] Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H.: Energy-efficient communication protocol for wireless micro sensor networks. In: Proceedings of the 33rd Annual Hawaii International Conference on System Sciences (January 2000)

[21] Wireless lan medium access control (mac) and physical layer (phy) specification, IEEE, New York, NY, USA, IEEE Std 802.11-1997 edition (1997)

[22] Intanagonwiwat, C., Govindan, R., Estrin, D.: Directed diffusion: a scalable and robust communication paradigm for sensor networks. In: Proc. ACM Mobi-Com, Boston, MA, USA, pp. 56–67 (2000)

Applications and Performances of Extended TTDDs in Large-Scale Wireless Sensor Networks

Hong Zhou[1], Lu Jin[1], Zhongwei Zhang[1], Hesham Ali[2], and Chulho Won[2]

[1] University of Southern Queensland, Toowoomba, Australia
{hzhou, jin, zhongwei}@usq.edu.au
[2] Univeristy of Nebraska, Omaha, U.S.A
{hali, cwon}@mail.unomaha.edu

Abstract. There are many applications of large scale sensor networks in which both the stimulus and the data collection stations are mobile (i.e. animal tracking, battlefield chasing). TTDD is a scalable and energy-efficient data dissemination model designed for this type of scenarios. However, TTDD only focused on handling mobile sinks. In this paper, we extend TTDD to deal with both mobile sources and sinks and evaluate the effectiveness of two potential extended TTDD schemes. Simulation results are presented to compare their performance in terms of energy consumption, delay and success rate. It is shown that the two schemes have similar performance when the moving speeds of the phenomena objects are slow. However, when the phenomena objects moving exceed certain speed, the advantages of one scheme over the other become unneglectable.

Keywords: sensor networks, data dessemination, mobile source, mobile sink, TTDD.

1 Introduction

The ultimate goal of communications is to communicate anything anywhere anytime. The recent advances of wireless mobile communications achieved in principle for people to communicate everywhere anytime, while the emerging wireless sensor networks add another dimension to communications by communicating anything. The anything in our lives can be as complicated and huge as few megabytes (or even gigabytes) video file or as simple and small as few bytes of any critical information. With the advances of microelectronics, the intelligent low-cost low-power small sensor nodes can be developed to sense almost anything which is of human's interest from the phenomenon of hazardous volcanoes and earthquake to the smell of the flowers at the garden. A sensor network consists of a large of number of sensor nodes which are densely deployed and connected through wireless links in a self-configured and self-organised manner. Such sensor networks would enable numerous new and exciting applications and bring another technology evolutionary wave to penetrate to every aspect of our lives (e.g. home, health, environment, military, agriculture, transport, manufactory, entertainment).

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 135–142, 2007.

However, the great convenience and functionality of sensor networks also bring significant challenges. A typical sensor has limited memory, power, and computational capacities; often sensor nodes are prone to failures and the topology of the network can dynamically change. Thus, several key issues such as resource constraints, unpredictability, large scale, real time and security must be addressed to enable the exploration of the full benefits of sensor networks. Due to the unique features of sensor networks, many protocols and algorithms proposed for traditional wireless ad hoc networks are not well suited to sensor networks. In addition, a sensor network is usually application-oriented. For different applications, there are different technical issues that need researchers to resolve.

There are many applications requiring large scale sensor networks where thousand or even tens of thousands of small sensors are distributed over a large geographic area to obtain fine-grained, high precision sensing data [1]. In such sensor networks, the sink(s) and source(s) can be either stationary or mobile. In this study, we adapt the same terminology in the research work addressed in [1], where a source is defined as a sensor node that detects a stimulus and report the stimulus, and a sink is defined as a user that collects the data reports from the sensor network. We are particularly interested in a large scale sensor network in which both the source and sink are mobile. This scenario often encountered whenever some tracking/chasing/searching and capture/find activities are involved. The accurate data about the mobile target(s)/source(s) has to be received by mobile sink(s) in a timely fashion. This category of sensor networks can be used for military battlefield, border protection, homeland security, and wildlife/pest control.

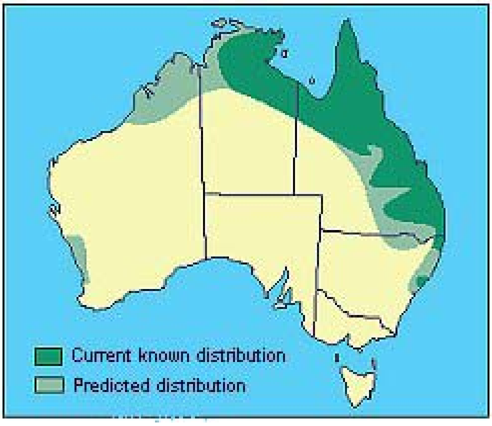


Fig. 1. Cane toad distributions in Australia (qouted from http://www.agric.wa.gov.au)

A potential application of such sensor networks is to control cane toads in Australia. Cane toads has been nominated as among 100 of the "world's worst" invaders by the invasive species specialist group of the World Conservation Union [2].The cane toad was introduced to Queensland, Australia to control pests of sugar cane in 1935. Since then, the cane toads adapt well into the Australian environment and the populations have exploded. The cane toads now have invaded into Northern

Territory and northern New South Wales. The main front is moving towards Western Australia (Refer to Figure 1). The natural rate of spread of Cane Toad is now 30-50 km/year in Northern Territory and about 5 km/year in New South Wales [2]. The toxic cane toads have significant agricultural, environmental, social, and culture impacts [3] and cause increasing problems in Australia. A sensor network deployed in an area where cane toads are populated will assist the environmentalists collect the cane toad movement information to monitor and control them from expanding aggressively.

The sink and source mobility imposes new challenges to sensor networks. The continuous movements of the source and sink require continuous location updates, which can lead to increased transmission collision and rapid power consumption. Although several data dissemination protocols [4, 5, 6, 7] have been developed for sensor networks, they do not perform efficiently for the applications with mobile sources and sinks. A scalable and efficient data dissemination model, called TTDD (Two-Tier Data Dissemination) has been proposed in [1] to address the problem. TTDD uses a grid structure so that only limited numbers of sensors (at the grid points) participate in the location updates. However, the model assumes that the sources are static and only sinks are mobile. The scenario with both mobile sinks and sources has not been studied and the problem with both mobile sinks and sources has not been fully explored. In this paper, we proposed two potential mechanisms to extend TTDD to accommodate the sensor networks with mobile sources and sinks, and we present the modification necessary to improve the energy consumption and performance of TTDD over the sensor networks, and then test this modified protocol on the simulated network

The rest of this paper is organised as follows. Section 2 briefly overviews data dissemination protocols and TTDD mechanism. Section 3 introduces the proposed extensions of TTDD to mobile sources. Section 4 describes the simulation model and presents the simulation results, analyses and compares the performance of the proposed schemes. Finally Section 5 summarises and concludes the paper.

2 An Overview of Two-Tier Data Dissemination (TTDD) Model

Energy-efficiency is one of the most important issues to be addressed in sensor networks. Several energy-efficient protocols have been proposed for delivering data to stationary or very low-mobility sinks (e.g. SPIN [3], DRP [4], GRAB [5]). However, TTDD is the first model which targets at efficient data dissemination to multiple mobile sinks in large-scale sensor networks. Each data source in TTDD proactively build a grid structure which enables mobile sinks to continuously receive data on the move by flooding queries within a local cell only.

We summarize the major principles of TTDD as follows:

2.1 Grid Construction

For each source, it builds a grid structure. The location of the source becomes the first cross-point of the grid. It then sends a data announcement message to each of its four adjacent crossing points and finally stops on the closest sensor node. The node stored the source information and further forwards the message to its adjacent nodes. Those nodes closest to the crossing locations are notified to become the dissemination node (DN). The process continues till a grid for the specific source is built.

It is assumed that the sensor field is a two-dimensional plane and divided into a grid of cells. The cell size is chosen as α and each cell is α × α square. Thus, for a source at location $L_s = (x, y)$, dissemination nodes are located at $L_p = (x_i, y_i)$ which are calculated as:

$$x_i = x + i \cdot \alpha,\ \ y_i = y + j \cdot \alpha\ \ (i, j = 0, \pm 1, \pm 2, \pm 3, \ldots) \tag{1}$$

2.2 Query and Data Forwarding

Once the grid is built and a sink needs data, it floods a query within a local area to discovery its intermediate dissemination node. The intermediate dissemination node forwards the query to the upstream dissemination node from which the intermediate dissemination node receives data announcements. The upstream one in turn forwards the query to its upstream one until finally the query reaches the source. During the above process, each dissemination node remembers its downstream dissemination node and later the data from the source is sent back to the sink along the way the query travels.

Once the data arrive at a sink's intermediate node, trajectory forwarding is used to relay the data to the sink. In trajectory forwarding, each sink communicates with the intermediate dissemination node through two sensor node agents: a primary agent and an intermediate agent. This mechanism enables the mobile sink to receive or send data from the source continuously. While the sink is constantly moving with unknown location, the intermediate dissemination node communicates with the sink through two stationary relaying agents. If a sink moves out the range of its current immediate agent, it picks another neighbouring node as its new immediate agent. Likewise, if the sink moves out of a cell from its primary agent, it picks a new primary agent and new immediate dissemination node.

2.3 Grid Maintenance

Each grid is set a Grid Lifetime at the time it is built. If the lifetime elapses and no update message received, the grid will no longer exists. To conserve the energy supply, TTDD does not periodically refresh the grid during its life time. The grid is maintained by on-going queries and upstream updates. TTDD trades computational complexity for less consumption of energy.

3 Extensions of TTDD (E-TTDDs)

In TTDD, it is assumed that once a stimulus appears, the sensor surrounding it collectively process the signal and one of them becomes the source to generate data reports. Then each source proactively builds a grid structure to relay the quires and data. Each source naturally becomes the first dissemination node of the grid. The studies in [1] are only focused on handling mobile sinks. The performance of different options in the scenario with mobile stimulus has not been addressed though the suggestions for mobile stimulus are briefly discussed in [1].

For comparison, we extend TTDD in two ways to accommodate the scenario with both mobile sources and sinks. The first approach is simply to rebuild the grids for each of the sources along the trail of the stimulus. For convenience of expression, we simply call this approach as E0-TTDD. This model may be suitable for stimulus which is not constantly moving or move slowly. If the stimulus is moving constantly, E0-TTDD needs continuously re-build the grids for each of the sources along the trail of the stimulus. The frequent grid constructions may increase the energy consumption significantly.

The second approach is to reuse the grid already built. When a source has a data to send, it floods a "Grid discovery" message within the scope of about a cell size to find an existing grid. For convenience of expression, we call this approach as E1-TTDD. In this approach, only the very first source node generates a data announcement message and disseminates it to create grid structure. Once the grid structure is completed, all subsequent source nodes send a query to search the closest DN within the scope of 1.3 α. if a valid DN is found, the source node may add the DN into its DN routing table and just utilize the existing grid structure to disseminate data. This approach will avoid redundant network traffic and save the scarce energy by decreasing the amount of active sensor nodes.

4 Simulation Model

Network simulation tool NS-2 is used for evaluation of the performance of two approaches. We implement the simulation model based on the original package provided by Ye [1]. The original model is developed in Ns-2.1b8a. It was modified and integrated to new ns2 version, namely Ns-2.29.

Similar to TTDD, we assume a square sensor field of area A in which N nodes are uniformly distributed. There are a number of N_k mobile sinks and a number of N_c mobile sources. The average moving speeds of sink and source are v_k and v_c, respectively. The sensor field is divided into cells with size of α by each source. The transmitting, receiving and idling power consumption rates of a sensor node are set to 0.66W, 0.395W, and 0.035W, respectively. The configuration parameters in the simulation are shown in Table 1.

We evaluate the impacts of different moving speeds on the performance of two extensions of TTDD (i.e. E0-TTDD and E1-TTDD). In the simulation settings, we choose different maximum speeds for the phenomena node (v_c =0, 5, 10, 15, 20 m/s). In this research, we limited our research on mobile source only as the two extensions have the same mechanism for dealing with mobile sinks.

Table 1. Simulation Parameters

Parameter	Value
Simulation time (s)	200
Number of sensor nodes	200
Area (m^2)	2000*2000
Source/Phenomena node moving speed (m/s)	0,5,10,15,20

5 Simulation Results

The three performance metrics considered in this research include energy consumption, packet delay, and success rate. The energy consumption is defined as the total accumulated transmitting and receiving energy of the participating sensor nodes consumed by the network. The idle energy is not counted for the purpose of performance comparison as it does not indicate the data delivery efficiency. The success rate is defined as the ratio of the total number of packets successfully received at the sink to the total number of packets generated at the source, average over all source-sink pairs [1]. The delay is defined as the average time difference between the instance at which the packet is generated at the source and the instance at which the packet is received at the sink.

We compare the performance of two TTDD extensions with different scenarios and parameters. In particular, we are interested in the impacts of the moving speed on the performance differences between E0-TTDD and E1-TTDD.

5.1 Impact of the Moving Speeds of Source Node

We first study the impact of the moving speed on the performance of two TTDD extensions. In this simulation scenario, we assume there are only one source object at one time and the moving speed of the phenomena object changes. Figure 1 shows the energy consumptions of E0-TTDD and E1-TTDD versus the moving speed. As the phenomena object moves faster, the energy consumption increases for both mechanisms. The faster a phenomena objects moves in a time slot, the more sensor nodes are activated as source nodes and start the data dissemination to DNs. However, the slope of the curve tends to decrease since the higher-tier grid forwarding changes only incrementally as sink moves. When the phenomena object moving speed below 5m/s, the energy consumptions for two schemes are almost same. However, when the phenomena object moves faster than 5m/s, it is shown that E1-TTDD consumed less energy than E0-TTDD. The energy saved by E1-TTDD is between 30% and 33.67% as the moving speed increases above 5m/s. E1-TTDD constructs and maintains grid

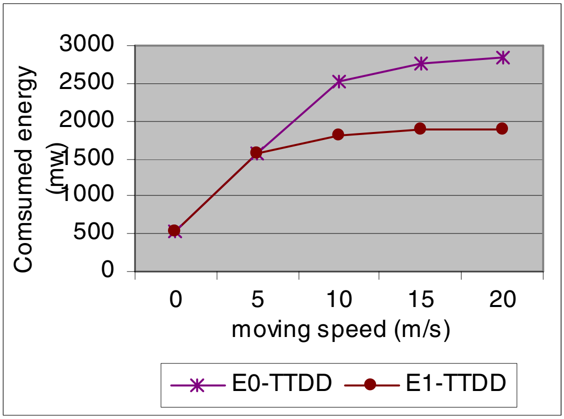


Fig. 2. Comparison of Energy Consumption

structure only once for the first source node and reuses the existing grid for all the subsequent activated source nodes. The faster the phenomena object moves, the more source nodes will be activated and thus the more energy will be saved by E1-TTDD.

Figure 3 compares the delay performance of the two schemes under the same simulation scenario. As the moving speed increases, the delay increases gradually for both protocols. However, E1-TTDD does not need to send flood packets to rebuild the grid thus shorter delay is experienced than E0-TTDD. The packets for rebuilding the grids increase the network traffic and thus increase the delay significantly.

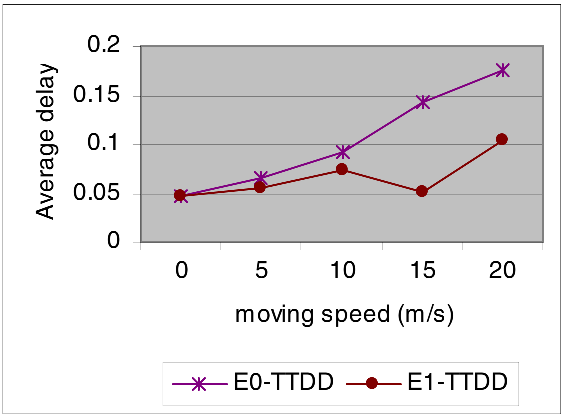


Fig. 3. Comparison of Average Delay

Figure 4 shows the successful rates of two protocols as the phenomena objects moving speed changes. As it is shown in Figure 4, the success rates of two protocols drops gradually from 1 to 0.7 as the moving speed increases, while the success rate of E1-TTDD is around 2.12% to 13.82% lower than E0-TTDD.

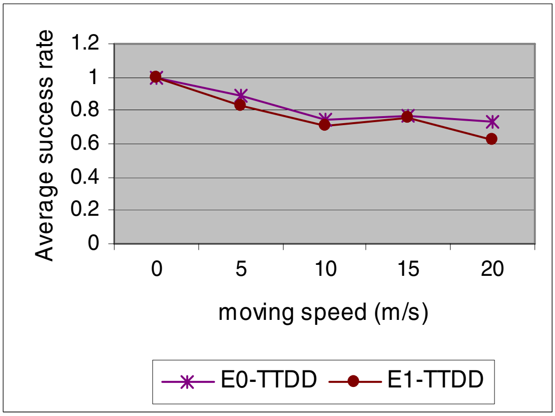


Fig. 4. Comparison of Average Success Rate

6 Summary and Conclusions

This paper presents some preliminary studies of two potential TTDD extensions (ie. E0-TTDD, E1-TTDD) in large-scale sensor networks where the stimulus and data sink are mobile. The two protocols distinguish from each other on the policy of building and maintaining a sensor grid. E1-TTDD reuses the grid built for the fisrt source node at the original location while E0-TTDD builds a new grid when new source node is activated along the track of the phenomena object. The simulation results show that their performance is similar if the moving speed is not very high (<5m/s). However, E1-TTDD has significant performance improvement than E0-TTDD when the stimulus moves at a relatively high speed (>5m/s).

The future research will investigate the impacts of number of active source-sink pairs on the performance of E-TTDDs, also the impacts of the size of the sensor network, the pattern of movements on their performance, and an optimized mechanism of handling mobile sources and sinks at different network scenarios.

References

1. Ye, F., Luo, H., Cheng, J., Lu, S., Zhang, L.: A Two-Tier Data Dissemination Model for Large-Scale Wireless Snesor Networks. In: ACM International Conference on Mobile Computing and Networking (2002)
2. Speare, R.: Bibliography of Bufo marinus, website http://www.jcu.edu.au/school/phtm/PHTM/staff/rsbufo.htm
3. Martin, G., Massam, M. : Cane Toads, Available from the website hosted by Department of Agriculture and Food, Western Australia http://www.agric.wa.gov.au
4. Intanagonwiwat, C., Govindan, R., Estrin, D.: Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. In: MOBICOM 2000. ACM International Conference on Mobile OCmputing and Networking (2000)
5. Coffin, D., Hook, D.V., McGarry, S., Kolek, S.: Declarative, Ad-Hoc Sensor Networking, SPIE Integrated Command ENviorments (2000)
6. Ye, F., Lu, S., Zhang, L.: GRAdient Broadcast: A Robust, Long Lived Large Sensor Network (2001), http://irl.cs.ucla.edu/papers/grab-tech-report.ps
7. Albowitz, J., Chen, A., Zhang, L.: Recursive Position Estimation in Sensor Networks. In: ICNP. IEEE International Conference on Network Protocols (2001)

Crosslayer Design Protocol Using Geographical Information in Wireless Sensor Networks

MyungJune Youn and Jaiyong Lee

Yonsei University, 134 Shinchon-Dong, Seodaemun-Gu, Seoul, Korea

Abstract. Many routing protocols for wireless sensor networks based on location information have been proposed such as GPSR, LAR, IGF, etc. When these routing protocols are used with existing MAC protocols, however, the sleep and wake-up scheduling of many MAC protocols degrades the routing protocol performance because of the long latency for periodic scheduling and synchronization overhead. In this paper, a cross layered routing and MAC protocol, CBT(Cell Back-off Time) protocol, has been proposed for the enhancement of successful routing in the Wireless Sensor Network environment considering the mobility of sensor nodes. Analysis and Simulation shows that CBT shows good performance when network density is high and nodes have mobility.

1 Introduction

Information gathered from sensor network becomes more meaningful when the data is combined with its location. For example, sensor network for temperature collection, temperature data itself may useful for some applications but such information is more practical when location of collected data is known. Then by using temperature and location information, it is possible to control temperature using air conditioner. Eventually most sensor networks should know its location to get more useful information. There are many researches to get sensor node's location, such as GPS or RSS. In this paper, we assume that all nodes in the network know their exact location. And most sensor networks have unique sink node which collects data. It means that a sink node is known to all sensor nodes. So we also assume that every nodes in sensor network know sink node's position.

There exist many protocols under these kinds of assumptions, for example, LAR[1] GPSR[2], IGF[3] etc. Most existing geographic based routing protocols uses periodic hello messages to gather location information of neighbor nodes. In order to follow continuously changing topology, however, all nodes should exchange hello message frequently. And it makes a lot of hello message overheads and degrades routing performance. IGF solves this problem by using back-off time. IGF chooses next hop dynamically, which enhances routing success ratio. But IGF uses 802.11 based MAC. It is not suitable for sensor network because sensor nodes need periodic sleep and wake up scheme in order to maximize network life time, which 802.11 is not suitable for such operation.

In this paper, we proposes a novel protocol, Cell Back-off Time(CBT) mapping protocol. CBT protocol do not need to know one hop neighbor node's location

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 143–152, 2007.

information like IGF and uses sleep and wake up scheduling efficiently by using crosslayer design.

2 Related Work

2.1 GPSR

GPSR(Greedy Perimeter Stateless Routing) is proposed for ad-hoc network. Basically GPSR uses greedy forwarding, which the closest to sink node is chosen as next hop node. Greedy forwarding is a basic algorithm in most geographical routing algorithm. The difference between algorithms is how to choose next node using minimum information about neighbor node. In GPSR, every node periodically exchanges hello message in order to update neighbor node position and make a decision of greedy forwarding. However, periodic hello message exchange wastes lots of wireless link resource and node energy.

2.2 IGF

IGF(Implicit Geographic Forwarding) is geographical routing protocol for sensor network. Basically IGF uses greedy forwarding like GPSR, but IGF des not need neighbor node information. IGF uses modified 802.11 DCF MAC. When source node have a packet to send, the source node broadcasts RTS to neighbor nodes in one hop range. All nodes received RTS calculate random back-off time and set back-off timer. When back-off timer is expired the node send CTS to the source node and other nodes will cancel back-off by overhearing. In order to forward packet to destination direction, IGF uses 60-degree sector as forwarding candidates.IGF is suitable for high density sensor network, because IGF does not need to manages neighbor nodes and also IGF decide next hop node dynamically when packet is forwarded. This algorithm makes its performance stable in high mobility. However, IGF uses modified 802.11 DCF MAC, which is not suitable for high density sensor network. In order to maximize sensor network life time it is necessary to use duty cycle. But 802.11 DCF MAC cannot support sleep and wake up scheduling efficiently.

3 CBT Protocol

In this section we present Cell Back-off Time(CBT) mapping protocol. CBT is proposed for sensor network, which topology changes dynamically. CBT uses back-off time like IGF, but CBT calculates back-off time by using cell. We will describe detailed definition of cell in the following subsection. CBT use cell not only for routing but also for MAC sleep and wake up scheduling. That is, the information(cell) used in routing need to be announced to MAC layer - crosslayer design.

3.1 Definition of Cell

Basically CBT protocol use back-off time to decide next hop node like IGF. In order to decide back-off time more effectively, CBT divide node's transmission

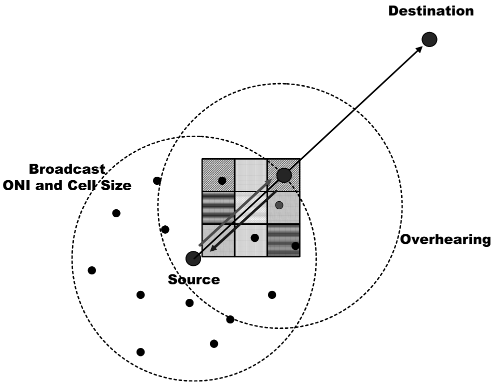


Fig. 1. CBT cell division and routing method

area into several cells. A Cell is a square type area, which side length of square is *cell_size*. In order to do cell dividing, CBT need one reference point first. We define that point as ONI[5]. It means Optimal Next-hop Information, which is the geographically closest location to the sink node in a source node's transmission range. This point is calculated by using source, sink node location and transmission range. ONI can be calculated by using internally dividing point formula. Let ONI position as (x_o, y_o), source node position as (x_s, y_s), sink node position (x_d, y_d) and node's transmission range as r. Then the ONI point is as following.

$$\begin{aligned} &d = \sqrt{(x_d - x_s)^2 + (y_d - y_s)^2} \\ &(x_o, y_o) = \left(\tfrac{r \cdot x_d + (d-r) \cdot x_s}{d}, \tfrac{r \cdot y_d + (d-r) \cdot y_s}{d}\right) \end{aligned} \tag{1}$$

ONI means geographically optimal position for shortest hop routing. However the ONI value can be changed using other parameters. For example, using node's energy, connectivity, mobility, etc. That means ONI can be used like a trajectory in TBF[4], which guides routing path using various parameters such as hop count, delay, connectivity, residual energy, etc. In this paper, this issue is excluded. We only focus on CBT's routing and MAC design using crosslayer design. We define a cell as *ONI_cell*, which is contains ONI location. *ONI_cell* is used for finding back-off time for routing. Detailed description is following in next subsection.

In order to do cell dividing, CBT also need *cell_size* which is determined by neighbor node density in one hop range. *Cell_size* is determined as possible as only a one node is exist in a cell. The reason for that is described in following subsection, CBT Routing Procedure. Neighbor node density is determined when network is initiated. Sink node flood its location information for a while when network is establish. During this network initiation procedure, each node collects sink node's location information and also tries to collect number of neighbor node. By using this procedure all nodes in the whole network maintain two value, sink node's position and *cell_size*.

With ONI and *cell_size* each node can divide transmission area into cells. These cells are used not only for routing but also MAC sleep and wake up scheduling. The detailed algorithms are described in each subsection.

3.2 CBT Routing Procedure

CBT routing is almost same to IGF except calculation of back-off time. When a node has a data to send, the source node calculates ONI and broadcasts ONI packet which contains ONI information to the neighbor nodes. The nodes listened the ONI packet calculate their *cell_position* which represent order of cell from *ONI_cell*. For example *cell_position* of *ONI_cell* is $(1, 1)$ and *cell_position* of just below the *ONI_cell* is $(1, 2)$. By using $cell\_position = (x, y)$ each node calculates their back-off time as following.

$$cell\_max = \frac{r}{\sqrt{2} \times cell\_size} \tag{2}$$

$$Back - off\ Slot$$

$$= \begin{cases} \sum_{i=0}^{x+y-2} (i) + |x-y| + u[\frac{x}{y} - 1] \\ \qquad\qquad\qquad\qquad \text{for } x + y - 2 \leq cell\_max \\ \sum_{i=0}^{x+y-2} i + \sum_{i=cell\_max}^{x+y-2} (2 \times cell\_max - i) + |x-y| + u[\frac{x}{y} - 1] \\ \qquad\qquad\qquad\qquad \text{for } x + y - 2 > cell\_max \end{cases} \tag{3}$$

Back-off time is calculated as Back-off Slot × *slot_time*. In equation (2) *cell_max* is defined for restrict forwarding area. If *cell_max* is set to infinite the forwarding area is almost same to transmission rage of node. In CBT *cell_max* is defined as equation (2) in order to restrict forwarding area only ±45-degree angle of the line connecting the source and sink node as shown in Figure 1. The reason that restrict the forwarding area is to minimize back-off node. By minimizing back-off nodes routing can be donewith minimum network resources. It lowers collision probability and increases routing success ratio.

After each node (received ONI packet) calculates their back-off time, nodes wait for back-off timer expired. When a node's back-off time is expired, the node replies to the source node. At this moment other nodes in back-off schedule, cancel their back-off timer. Then only a one node will respond to source node and the node is selected as a next hop node. By repeat this procedure routing path from a source to the sink is established. When there is no response from neighbor node because there is no nodes in forwarding area or collision occurs, the source node can route other way by changing its ONI value or *cell_size*. When collision is detected in MAC layer, that means two or more nodes are exist in a same cell. In this case, the source node can reduce *cell_size* and avoid collision. When there is no response because there are no nodes in forwarding area, the source node can rotate ONI value some degree and goes around using other path.

3.3 CBT MAC Sleep and Wake Up Procedure

In this section, we describe a sleep and wake up scheduling method in MAC protocol, which uses location information of sensor node. In order to send packets with low latency, sensor networks have to wake up all the time. However, sensor nodes have limited energy so all nodes should have periodic sleep and wake up scheduling to maximize network life time. This kind of mechanism increase network latency. In most sensor MAC protocols sleep and wake up schedule have to be synchronized through whole network or localized network because communication between nodes is possible only when nodes are wake up. However CBT do not need to synchronize whole network sleep and wake up time. It can be achieved by using location information of node - cells made during routing procedure. When a source node sends ONI packet, neighbor nodes calculate their *cell_postion* and decide back-off time. During this procedure each node decide its MAC scheduling too. Figure 2 shows sleep scheduling in CBT. As shown in the figure, each node's scheduling is determined by *cell_position*, some cells go into sleep state and others stay wake up. By using this method, there always exist wake up nodes in network so latency caused by MAC can be minimized. Also there is no need to synchronize through out whole network. Nodes just follows one-hop sleep and wake up schedule and each hop is independently scheduled. This is possible because there always exist wake up nodes in each hop. Policy for scheduling is just follow ONI packet sender's schedule and when a node receives other ONI packet during the sleep and wake up schedule, ignores that schedule. When a node did not received ONI packet for a while so there is no schedule, the node randomly generate dummy ONI packet and start self scheduling.

In left of figure 2, duty cycle of a node is 50%. Change of duty cycle is simple. Just sub-grouping sleep and wake up nodes. Right of Figure 2 is sub-grouping four nodes and reduce duty cycle to 25%. By using this method duty cycle can be dynamically changed. However, number of cell is determined by neighbor node density in one hop range. So there may not exist enough cells satisfying duty cycle. In that case, hybrid sleep and wake up schedule can be used. By using traditional scheduling and CBT scheduling at the same time, not only latency but also energy efficiency can be achieved together. And CBT MAC reduces

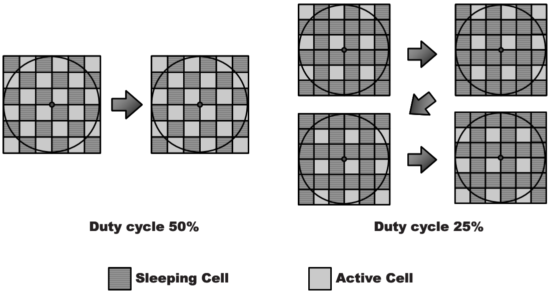


Fig. 2. CBT sleep and wake up scheduling

collision probability because number of node in transmission range is restricted by sleep and wake up schedule. It makes more efficient communication between nodes. It may seems that CBT scheduling consumes more energy than periodic sleep and wake up schedule, but both use same energy. It is proved in analysis.

4 Analysis

4.1 Routing

In this section we will define a environment when CBT protocol operates efficiently compared to existing geographical routing algorithm, GPSR. GPSR shows more low latency than CBT because CBT uses back-off time in order to decide next hop node. But GPSR can decide next hop instantly because GPSR knows its one hop neighbor node by using periodic hello message. However periodic hollo message exchange seriously affects to routing when nodes have mobility. So we will compare control packet overhead which is used packet's size during routing and delay parameters of CBT and GPSR and define a environment when CBT works more efficiently work.

First, GPSR overhead is defined as following.

$$OH_{GPSR} = \frac{pk\_size \times E[n] \times E[h]}{E[t]} \tag{4}$$

In equation (4) $E[n]$ is average number of neighbor node. $E[h]$ is average hop count from a source to the sink, $E[t]$ is hello message period for neighbor node information(This value is inverse proportional to node mobility. When node's mobility is high, hello message exchange period should be short). Because GPSR periodically exchanges hello messages with all neighbor nodes, OH_{GPSR} can be defined to equation (4). And CBT routing overhead is as following.

$$OH_{CBT} = 2 \times pk\_size \times E[h] \times E[d] \tag{5}$$

In equation (5) $E[d]$ is average data rate. Because CBT need two control packets for routing, a ONI packet and a response packet, OH_{CBT} is linear with $2 \times pk\_size$. And CBT is on-demand routing protocol. So routing overhead can be defined as equation (5). But routing overhead can be reduced by using routing cache. In this analysis we will assume worst case, always reroutes when a packet arrives.

Next, CBT has additional delays caused by back-off time. This can be represented as expected time slot. So it can be represented as following([$E[cell]$ can be calculated by simple urn and ball probability problem. It is omitted because of paper limitation).

$$Delay_{CBT} = E[cell] \times time\_slot \tag{6}$$

By using equation (4), (5), (6) efficiency ratio is as following.

$$\begin{aligned} Efficiency &= \frac{OH\_CBT \times Delay\_CBT}{OH\_GPSR} \\ &= \frac{2 \times time\_slot \times E[d]E[t]E[cell]}{E[n]} \end{aligned} \tag{7}$$

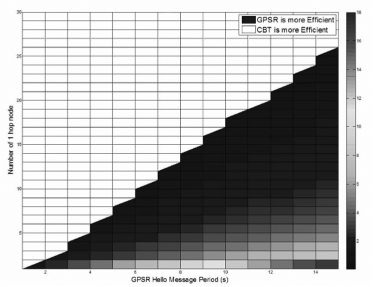


Fig. 3. Efficiency compare GPSR and CBT

Equation (7) is basis of whether CBT is efficient or inefficient compared to GPSR. If equation (7) is smaller than one, it means CBT is more efficient. The result is shown in Figure 3. Dark area represents GPSR is more efficient and white area represents CBT is more efficient. In the figure, x-axis is GPSR hello message period, inverse of node's mobility. As x-axis value increases, mobility goes lower, area that GPSR is more efficient increases. That means GPSR operates efficiently in static network. And y-axis is neighbor node number. So in dense network CBT works more efficiently. From the Figure 3, CBT is suitable for high density and high mobility, which means CBT is suitable for dynamically changing sensor network.

4.2 MAC

In this section we analysis energy consumption of CBT MAC and periodic sleep and wake MAC. We will calculate one hop node's energy consumption. First, we define a node's energy consumption as following.

$$\begin{aligned} E_{node} &= E_{sleep} + E_{idle} + E_{tx} \\ &= P_{sleep}T_{sleep} + P_{idle}T_{idle} + P_{tx}T_{tx} \end{aligned} \tag{8}$$

In equation (8) P is power and T is time for the each state. For a periodic sleep and wake up schedule, time of each state is as following.

$$\begin{aligned} T &= T_{wake} + T_{sleep} \\ T_{sleep} &= T - T_{wake} \\ T_{wake} &= T_{idle} + T_{tx} = T_{idle} + pT_{wake} \\ T_{idle} &= T_{wake} - T_{tx} = (1-p)T_{wake} \end{aligned} \tag{9}$$

In equation (9) p is percentage of transmission time during wake up state. Substitute equation (9) to (8) and simplify.

$$\begin{aligned} &E_{node-periodic} \\ &= P_{sleep}T + [P_{rx} - P_{sleep} + p(P_{tx} - P_{rx})]T_{wake} \end{aligned} \tag{10}$$

In order to get energy consumption per time, divide equation (10) by T then,

$$\begin{aligned}&E_{node-periodic-per-time}\\&= P_{sleep} + [P_{rx} - P_{sleep} + p(P_{tx} - P_{rx})]\tfrac{T_{wake}}{T}\\&= P_{sleep} + [P_{rx} - P_{sleep} + p(P_{tx} - P_{rx})]D\end{aligned} \tag{11}$$

where D is duty cycle,

$$D = \frac{T_{wake}}{T_{wake} + T_{sleep}} \tag{12}$$

For a CBT MAC sleep and wake up schedule time of each state is defined as following.

$$\begin{aligned}T_{wake} &= T\\T_{sleep} &= \tfrac{1-D}{D}T\\T_{tx} &= pT_{wake} = pT\\T_{idle} &= T_{wake} - T_{tx} = (1-p)T\end{aligned} \tag{13}$$

Substitute equation (13) to (8) and simplify.

$$\begin{aligned}&E_{node-CBT}\\&= [P_{sleep}\tfrac{1-D}{D} + (1-p)P_{rx} + pP_{tx}]T\end{aligned} \tag{14}$$

In order to get energy consumption per time, divide equation (14) by $T_{wake} + T_{sleep} = T/D$ then,

$$\begin{aligned}&E_{node-CBT-per-time}\\&= [P_{sleep}\tfrac{1-D}{D} + (1-p)P_{rx} + pP_{tx}]D\\&= P_{sleep}(1-D) + D(1-p)P_{rx} + pDP_{tx}\\&= P_{sleep} + [P_{rx} - P_{sleep} + p(P_{tx} - P_{rx})]D\end{aligned} \tag{15}$$

Equation (11) and (15) is same which means that CBT MAC scheduling consumes same energy with periodic sleep and wake up scheduling.

5 Simulation

To evaluate performance, CBT is implemented by using OPNET 11.0. Simulation parameters are listed in Table 1. Simulation is done by using square topology with unique sink node. And mobility model is generated by OPNET random mobility. Simulation result is compared to GPSR.

Table 1. Simulation Parameters

Parameters	Settings
Network Topology	300(m) x 300(m)
Node Number	500
Transmission Range	30(m) - 60(m)
Traffic Model	exp. with 10(s)
Mobility Model	uniform 0 - 10 (m/s)

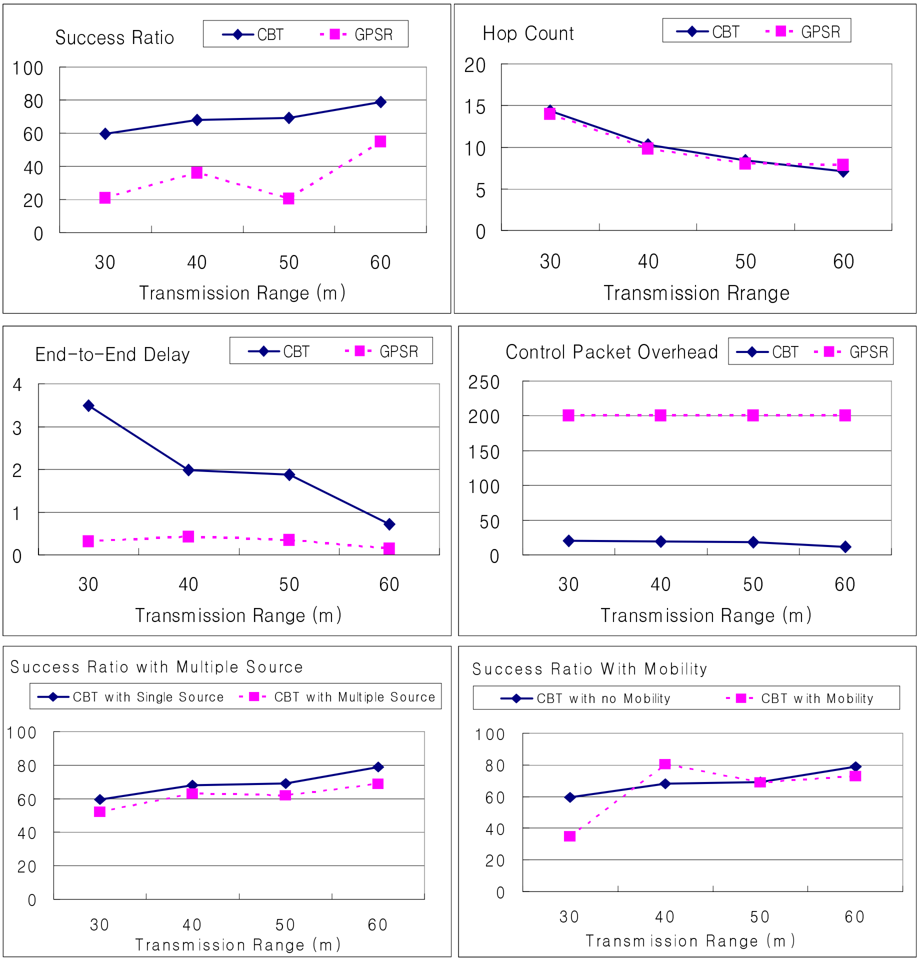


Fig. 4. Simulation result

In Figure 4 GPSR shows low routing success ratio. GPSR broadcasts hello messages periodically which causes collision with data packets. However, CBT shows 20 - 40% higher success ratio than GPSR. it is because CBT uses back-off time without exchanges hello message. Also average hop count from source to sink is almost same with GPSR. But control packet overhead is much lower than GPSR. GPSR's control packet overhead is determined by hello message period so if we set long hello message exchange period control packets will be decreased. But GPSR shows const control packet overhead even the topology changes. On the other hand, CBT's control packet overhead is changes as topology changes, which means that CBT adapts to the network topology and generate only required control packets. However, CBT has long latency compared to GPSR. It is natural because CBT uses back-off time in order to decide next hop. But as shown in Figure 4 latency difference becomes smaller as transmission range increases, or density of one hop neighbor node increases. And the latency is reasonable in high density network. CBT works well even in multiple source or nodes have a mobility. As shown in the Figure 4, success ratio with multiple source shows almost same performance. Slight drop of success ratio is caused by collision of ONI packets but it is not so significant. And when nodes have mobility success ratio is not dropped. Because CBT dynamically decide next hop node when packet arrives, mobility does not affect much to routing. But when number of one hop neighbor node is not sufficient, routing success ratio may fall down.

6 Conclusion

Existing location based routing protocols have compatibility problem with periodically sleep and wake up scheduling MAC. In this paper, we proposed a novel protocol CBT - Cell Back-off Time mapping, which uses the cross-layer design method in order to optimize routing and MAC sleep scheduling. CBT dynamically decides the next hop node using cell and back-off time which minimizes routing overhead and suitable for frequent topology changes of the sensor network. And by using the cross-layer design, the latency caused by MAC sleep and wake up scheduling is minimized. Also this scheduling is localized only in one hop range so synchronization over whole network is unnecessary.

Acknowledgement

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) IITA-2006-C1090-0603-0038

References

1. Ko, Y.-B., Vaidya, N.H.: Location-aided routing (LAR) in mobile ad hoc networks. In: International Conference on Mobile Computing and Networking. In: Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking, pp. 66–75
2. Karp, B., Kung, H.: GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. In: MOBICOM 2000. ACM International Conference on Mobile Computing and Networking (2000)
3. He, T., Blum, B.M., Cao, Q., Stankovic, J.A., Son, S.H., Abdelzaher, T.F.: Robust and timely communication over highly dynamic sensor networks. Real-Time Systems Journal, Special Issue on Real-Time Wireless Sensor Networks (2007)
4. Niculescu, D., Nath, B.: Trajectory based forwarding and its application. In: MOBICOM 2003. ACM International Conference on Mobile Computing and Networking (September 2003)
5. Lee, J.: Energy Efficient Geographical Time Back off Routing for Wireless Sensor Networks. In: RFID/USN Korea 2005 International Conference, Seoul, Korea, Track 5, Session 1 (October 14, 2005)

An Efficient Data Exchange Protocol Using Improved Star Trees in Wireless Sensor Networks

Ben Xu[1], Liusheng Huang[2], Hongli Xu[3], Jichun Wang[4], and Yang Wang[5]

Depart. of Computer Science and Technology, Univ. of Science & Technology of China
Anhui Province key Laboratory of Software in Computing and Communication
Hefei 230027, P.R. China
{ustrenyy[1], hlxu3[3], jichunw[4], angyan[5]}@mail.ustc.edu.cn,
lshuang[2]@ustc.edu.cn

Abstract. In wireless sensor networks, it is necessary and important to send information to all nodes. In some situation, every node has its own data to send to all the other nodes. The communication patterns are all-to-all broadcasting, which is called data exchange problem. In this paper, we present an efficient data exchange protocol using improved star trees. We divide the sensor area into four equal grids and each sensor node associates itself with a virtual grid based on its location information. These grids can be divided again if necessary. In each grid, we calculate the position of root node with location information of sensor nodes. Then, an efficient data exchange Star-Tree was constructed and used to achieve the exchange behavior in the grid. The fused data of each grid was sent to the center node. Simulations show that our protocol can prolong the lifetime about 69% to the multiple-chain protocols, and the delay can be reduced at least 35%.

Keywords: Data Exchange, Sensor Networks.

1 Introduction

Recent advances in microelectronic technology have enabled the production of compact and inexpensive wireless sensors. Large numbers of sensor nodes can be deployed in an appointed area to form wireless sensor networks [6,9]. Wireless sensor networks have been used for numerous applications including military surveillance, facility monitoring, and environmental monitoring. Periodically, sensor nodes gather data from surroundings and transmit sensed data to destination for processing. These sensor nodes are equipped with constrained energy and limited computing capabilities. Therefore, energy is a scarce resource that must be conserved to the extent possible in sensor networks.

Much research effort has been made on data gathering (or aggregation) [5] and data dissemination [2] protocols for wireless sensor networks. A cluster-based data gathering protocol, called LEACH, is proposed in [4]. In PEGASIS [3], a linear-chain scheme is used to gather data in sensor networks. The author in paper [8] studies the transport capacity of many-to-one communication in data gathering, which is achieved by a hierarchical architecture. SAFE, in paper [2], attempts to save energy

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 153–164, 2007.

through data dissemination path sharing among multiple data sinks. A modified flooding, called LAF [7], is used to disseminate data in wireless sensor networks.

However, the communication models are either one-to-many or many-to-one. In data gathering, the base node is the destination and all the other nodes are the sources. While in data dissemination, the multiple sinks are the destinations. A similar but different case is all-to-all broadcasting, which is termed as data exchange. In data exchange, every node has to send its information to all the other nodes. So each node is both the source node and the destination node. Kemei Du et al. proposed a multiple-chain protocol for all-to-all broadcasting in [1].

Little attention was paid on data exchange. Data exchange can be used in applications such as SSS (Single-Soldier System). GPS equipments are widely used in SSS, so every soldier in the battlefield will get its accurate location information, which will be sent to all the other soldiers by data exchange. Besides, with biologic sensors, vital sign data of soldiers can be transmitted to everyone. If anybody is injured in the battlefield, the nearest soldier can give him first aid as soon as possible.

In this paper, we present an efficient data exchange protocol using improved star trees, called DEIST (Data Exchange with Improved Star Trees). DEIST has two goals: prolonged lifetime and reduced delay of the networks. The sensing area is divided into four equal grids. For the purpose of energy conservation, an improved star tree is constructed in each grid. We adopt the idea of the incremental MST algorithm, where one node is added to the tree in each round. Local grid information is gathered by the tree, and then it is sent to the global data center node of the whole sensing area. The center node aggregates the received packets and disseminates them to four grids.

Our DEIST efficiently reduced the longest length of the communication links. The maximum number of hops from nodes to center node is also decreased. Besides, the problem in [1] that ending node of the chain protocols is too far from the center node is solved in our DEIST, we have several candidate gateway nodes. Therefore, our scheme outperforms the existing multiple-chain scheme.

The remainder of the paper is organized as follows. Section 2 introduces basic notations and assumptions. A detailed description of the proposed scheme is presented in Section 3. Simulation study is conducted in Section 4. Finally, in Section 5 we conclude the paper.

2 Preliminaries

This section presents the basic model that our DEIST takes in data exchange, where each sensor node will receive messages from all the other sensors. We consider a large number of wireless sensor nodes deployed in a flat geographical space. Each sensor node is assumed to be aware of its own location by the positioning equipment, or the positioning algorithms. Once deployed, sensor nodes will not leave its position. Sensor is powered by its battery, whenever a sensor transmits or receives a data packet it consumes some energy from its battery. When energy is gone, the node turns to dead and becomes useless. Each sensor node produces some information periodically as it monitors its vicinity. We assume that each sensor generates one data packet per time unit, and we refer to call each time unit as a round. Every round, the packet should be sent to all the other sensor nodes for processing. We improve the communication model presented by Heinzelman et al. [4]. It assumes that each sensor has the ability to transmit its packet to any other sensor in the network. In practice, some sensor nodes communicate with other nodes across multiply hops, they can not communicate with each other directly. So in our model, each sensor can merely communicate with those sensor nodes within its maximum communication range, and wireless channels are bidirectional.

The first order radio model described in [4] was used for our energy consumption. In this model, a radio consumes $E_{elec} = 50nJ/bit$ to run the transmitter or receiver circuitry and $E_{amp} = 100pJ/bit/m^2$ for the transmitter amplifier. The radios have power control and can adjust their power level to reach the appointed sensor node with the minimum energy cost. The equations that are used to calculate transmission costs and receiving costs for $k-bit$ message and a distance d are shown as below:

$$\textbf{Transmitting}\quad E_{Tx}(k,d) = E_{elec} * k + E_{amp} * k * d^2 \tag{1}$$

$$\textbf{Receiving}\quad E_{Rx}(k) = E_{elec} * k \tag{2}$$

A fixed amount of energy is spent in receiving and transmitting a packet in the electronics, and an additional amount proportional to d^2 is spent while transmitting a packet.

In [5] Data aggregation is defined as the process of aggregation the data from multiple sensors to eliminate redundant transmission and provide fused information to the target node. Data aggregation attempts to collect the most critical data from the sensors and make it available to the sink in an energy efficient manner with minimum data latency. It is an efficient way to save energies in wireless sensor networks and has been widely used in previous works [10,11]. In our paper, we also make the assumption that an intermediate sensor can aggregate multiple incoming packets into a single outgoing packet.

3 Efficient Data Exchange Protocol Using Improved Star Tree

Our DEIST protocol is described in this section. With the help of improved Star-Tree, we succeed in data aggregation of each grid. Then, the fused data is disseminated to other grids by the global center node.

3.1 Star-Tree Definition

Wireless sensor network is represented as an undirected connected graph $G = (V, E)$, where V is the set of n nodes, and E is the set of edges representing the direct communication between the sensors within the sensor communication range.

Chain-based protocols in [1] try to construct a low-energy-consumption transmission chain in target area to solve the all-to-all data exchange problem. As we know that the disadvantages of chains lead to the long delay in data packets transmission and the large links between nodes. Given a connected graph $G = (V, E)$, using the square of distance between two nodes as the weight of edges in $G = (V, E)$, its MST (Minimum Spanning Tree) has the Minimum total weight. Although MST is excellent in the total weight of the spanning tree, it is not fit for all-to-all data exchange. There is too much data that should be stored in intermediate nodes, so we try to use a new structure Star-Tree for the local sensor network.

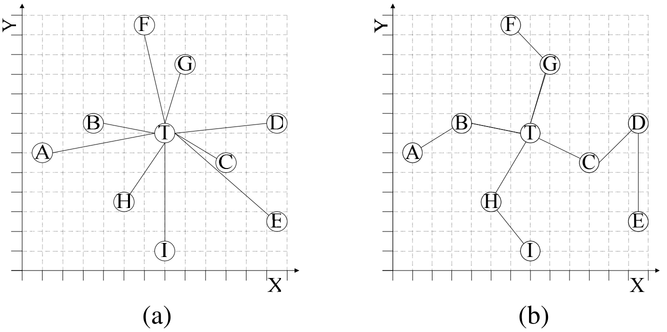


Fig. 1. (a) Illustration of Primary Star-Tree (b) Illustration of Improved Star-Tree

Definition 1. *Primary Star-Tree*
Given a tree $T=(V_T,E_T)$ with a root point $t\in V_T$, $D(u)$ is used to describe the connectivity of point u, for $\forall u\in V_T$. It is evident that $D(u)\geq 1$ for tree T is a connected graph.

If $$D(t)\geq 1 \quad t\in V_T$$
$$D(u)=1 \quad u\in V_T \ \ and \ \ u\neq t$$

Then the tree T is called as a Primary Star-Tree T as shown in Fig 1 (a).

3.2 Root of Star-Tree Selection Mechanism

The root of Star-Tree is the data center of local grid. The performance of Star-Tree will greatly be influenced by the position of the root. In Fig 1 (a), if node A was selected as the root node of the tree, the energy cost of the spanning tree is much more than that of node T. In other words, it is very important to find the position of the root node. In our DEIST, the root is centered at the node that is closest to the Vertex-Barycenter of the local network.

Definition 2. *Vertex-Barycenter*
Let $V(n)=\{A_1,A_2,...,A_n\}$ be a set of n points in the Euclidean plane. Let $\overrightarrow{A_1},\overrightarrow{A_2},...,\overrightarrow{A_n}$ be the location vectors for points $A_1,A_2,...,A_n$, respectively. We term $\vec{G}=\frac{1}{n}\sum_{k=1}^{n}\overrightarrow{A_k}$ as the Vertex-Barycenter of point set $V(n)$.

Lemma 1. *We assume G as the Vertex-Barycenter of point set $V(n)$. For any point P in the Euclidean plane, the following equation is correct.*

$$\sum_{k=1}^{n}\left|\overrightarrow{PA_k}\right|^2 = n\left|\overrightarrow{PG}\right|^2 + \sum_{k=1}^{n}\left|\overrightarrow{GA_k}\right|^2 \tag{3}$$

Proof
First, equation $\overrightarrow{PA_k} = \overrightarrow{PG} + \overrightarrow{GA_k}$ can be achieved according to vector algorithm.

$$\left|\overrightarrow{PA_k}\right|^2 = \left|\overrightarrow{PG}\right|^2 + \left|\overrightarrow{GA_k}\right|^2 + 2\overrightarrow{PG}\cdot\overrightarrow{GA_k}$$

$$\because \sum_{k=1}^{n}\overrightarrow{GA_k} = 0$$

$$\therefore \sum_{k=1}^{n}\left|\overrightarrow{PA_k}\right|^2$$

$$= n\left|\overrightarrow{PG}\right|^2 + \sum_{k=1}^{n}\left|\overrightarrow{GA_k}\right|^2 + 2\overrightarrow{PG}\cdot\sum_{k=1}^{n}\overrightarrow{GA_k}$$

$$= n\left|\overrightarrow{PG}\right|^2 + \sum_{k=1}^{n}\left|\overrightarrow{GA_k}\right|^2$$

End of Proof

Theorem 2. $\left|\overrightarrow{PA_k}\right|$ *denotes the distance between point P and point* A_k. *G is the right point to minimize the value of* $\sum_{k=1}^{n}\left|\overrightarrow{PA_k}\right|^2$.

Proof
For any point P in the Euclidean plane, the conclusion of Lemma 1 can be used to get the following equation:

$$\sum_{k=1}^{n}\left|\overrightarrow{PA_k}\right|^2 = n\left|\overrightarrow{PG}\right|^2 + \sum_{k=1}^{n}\left|\overrightarrow{GA_k}\right|^2 \cdot$$

The former part $n\left|\overrightarrow{PG}\right|^2$ is a variable while the latter part $\sum_{k=1}^{n}\left|\overrightarrow{GA_k}\right|^2$ is a constant.

$$\therefore \sum_{k=1}^{n}\left|\overrightarrow{PA_k}\right|^2 = n\left|\overrightarrow{PG}\right|^2 + \sum_{k=1}^{n}\left|\overrightarrow{GA_k}\right|^2 \geq \sum_{k=1}^{n}\left|\overrightarrow{GA_k}\right|^2 \tag{4}$$

The mark of equality establishes if and only if $P = G$.
End of Proof

According to our energy consumption model described in section 3, a fixed amount of energy and an additional amount proportional to d^2 are spent in transmitting a packet over the distance d. Hence, we conclude that G is the perfect point to be the local data center of each grid. In order to deal with the problem that point G refers to a nonexistent node in the sensor networks, we choose the node which is the closest to G as the root of spanning Star-Tree.

To calculate point G, we should gather the location information of all the other sensor nodes to one node. The Primary Star-Tree can be used for the first time to get G. First, select the sensor node which is the closest to the geographical center of the local grid as the root of Primary Star-Tree. Next, all the other sensor nodes will send their location data to the root directly. Third, G is calculated in root node. At last, the node which is the closest to G as the root of spanning Star-Tree. We can get the conclusion from the context that the complexity to calculate the root is $O(n)$.

As we mentioned in abstract that virtual grid is used in our protocol, the location information of local grid is enough to calculate G in each grid, so our protocol can be implemented in a distributed manner.

3.3 Improved Star-Tree(IST) Construction

The power is the most significant resource in wireless sensor network. Energy conservation is an very important factor to be considered in designing the protocol for sensor networks. With the current technology, long distance transmission in wireless sensor networks is very costly. Although, the structure of primary Star-Tree is fit for all-to-all data exchange, it is not a good choice. The amount of energy consumed in transmission is too much for each sensor node sending its packet directly to the root node. The primary Star-Tree is not energy-efficient enough to be adopted. Decreasing distance is an effective step to prolong the lifetime of the network. As a result, we improve on the primary Star-Tree to propose a new Star-Tree. The definition of the improved Star-Tree is shown as follows.

Definition 3. *Improved Star-Tree*
Given a tree $T=(V_T,E_T)$ with a root point $t \in V_T$, $D(u)$ is used to describe the connectivity of point u, for $\forall u \in V_T$. It is evident that $D(u) \geq 1$ for tree T is a connected graph.

If $D(t) \geq 1 \quad t \in V_T$

$2 \geq D(u) \geq 1 \quad u \in V_T \;\; and \;\; u \neq t$

Then the tree T is called as an Improved Star-Tree T, as shown in Fig 1 (b).

The IST construction algorithm presented below shows us how to build an improved Star-Tree on a given graph. First, the algorithm computes the *Vertex-Barycenter* of the graph, whose properties have been proved to be efficient for data exchange in the context. Second, the sensor node which is the closest to the *Vertex-Barycenter* is chosen as the root of our improved Star-Tree. The third step is to construct the data exchange tree with the sensor node calculated by the first step. We adopt the idea greedy algorithm, where one node is added to the spanning tree in each round.

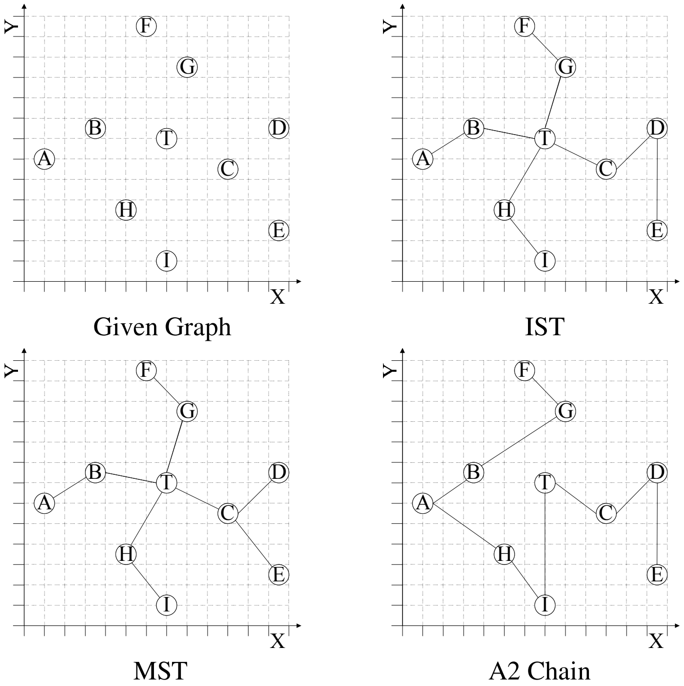
Assume that T_{left} is the set of the points that have not joined the tree, and T is the set of points that have been in the spanning tree. T_{active}, the sub set of T, is composed of node t and the nodes whose connectivity is equal to one. At first, only the root

```
IST Construction Algorithm:
Compute the Vertex-Barycenter $s_0$
Select the node $t$ that is the closest to $s_0$
$T \leftarrow \{t\}$;
$T_{active} \leftarrow \{t\}$;
$T_{left} = V$ ; // $V$ is the set of nodes in the grid
while $T_{left} \neq \phi$ do
    $\forall u \in T_{left}$ , $\forall v \in T_{active}$ ,
    choose $u$ with the $Min[d(u,v)]$
    $T_{left} \leftarrow T_{left} - \{u\}$ ;
    $T \leftarrow T \cup \{u\}$;
    $u$ choose $v$ as its parent in $T$ ;
    Update $T_{active}$ based on the current $T$ ;
end-while
```

Fig. 2. IST Construction Algorithm

node, labeled as t, is in the tree, and all the other nodes become the candidate nodes. In each round, we select a pair of nodes u ($u \in T_{left}$) and v ($v \in T_{active}$), with the minimum distance $d(u,v)$. Then, node u is deleted from set T_{left}, and accepted by Star-Tree across node v. In the spanning tree, node v is called the parent of node u, and u is the child of v. As the current tree is changed, set T_{active} should be updated in the end of each round. The algorithm is terminated until all the points in set T_{left} have joined the Star-Tree. The formal description of the algorithm is given in Fig. 2.

The following tables in Fig. 3 describe the situation regarding attributes for IST, MST and A2 Chain [1]. MST has the best weight ($\sum d^2$), which is about 105.5 units. IST and A2 have biggish datum which are 115.25 and 160.75 units respectively. The weight concerns the energy consumption very nearly. So the amount is the less the better. Delay is another important performance measure for wireless sensor network. Given different graphs with N nodes, the depth of MST and IST will vary from 1 to (N-1) hops, A2 has the longest depth which is always N-1 hops. The complexity of MST is $O(n^2)$, while A2 is $O(n^3)$ which has been proved in [1]. Now we analyze the time complexity of our IST algorithm. In the first step, there are $O(n)$ intersection points to calculate the *Vertex-Barycenter*. And the time for making sure the root is not more than $O(n)$. For the last step, it needs time $O(n^2)$ to compute the spanning tree. Hence, the total time complexity for constructing the IST is about $O(n^2)$.



Given Graph

IST

MST

A2 Chain

	IST	MST	A2
Weight$\left(\sum d^2\right)$	Middle	Best	Worst
Hop	1 ~ (N-1)	1 ~ (N-1)	N-1
Complexity	$O\left(N^2\right)$	$O\left(N^2\right)$	$O\left(N^3\right)$

Fig. 3. Comparisons over IST, MST and A2 Chain

3.4 Data Exchange Using IST

Our sensing area is divided into four equal grids. The global data center of the network is located at the node which is the closest to the center of the sensing area. And the global center node does not belong to any grid. In each grid, an IST is built to take over the data exchange task. The root of the IST is the local data center of each grid. In each grid, a gateway node is selected to communicate with the global data center directly. The leaf node which is the nearest to the global center is elected to be the gateway. It is illustrated in Fig. 4 that node E is the gateway of grid 1. The gateway will report to the root node along the sub chain of the IST from gateway to root. The receiver of this report message will choose the sender as its new parent. So root T chooses C as its parent.

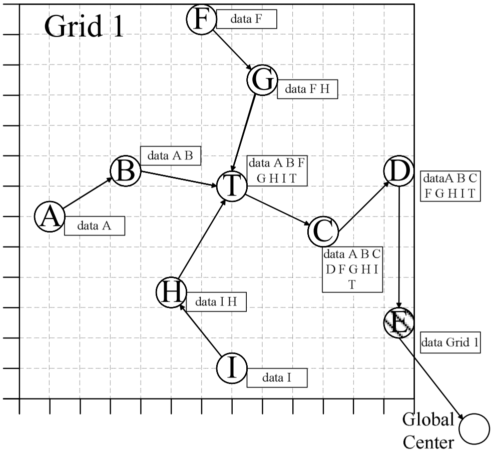


Fig. 4. Illustration of data aggregation process

- **Data Aggregation Process**

Each round, leaf nodes start to transmit packet to their parents except the gateway. Relaying nodes add local data into the received packet. No packets received, no packets sent. Once root having gathered all the packets of its children (without node C), an aggregated packet is sent to its parent. It is evident that the packet sent by the gateway E to center contains all the information of grid 1. The illustration of data aggregation process is shown in Fig. 4.

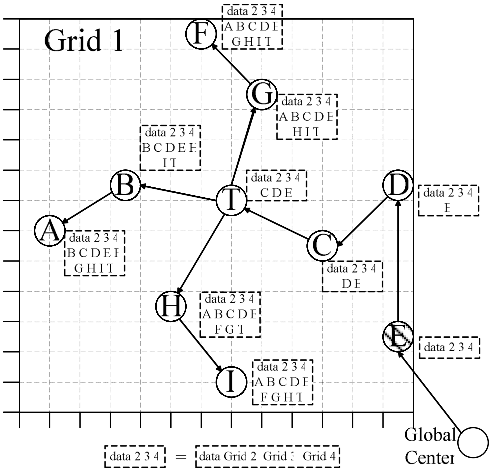

Grid 1
F
data 2 3 4 A B C D E G H I T
G
data 2 3 4 A B C D E H I T
data 2 3 4 B C D E F I T
B
T
data 2 3 4 C D E
D
data 2 3 4 E
A
data 2 3 4 B C D E F G H I T
C
data 2 3 4 D E
H
data 2 3 4 A B C D E F G I
E
data 2 3 4
I
data 2 3 4 A B C D E F G H I
Global Center
data 2 3 4 = data Grid 2 Grid 3 Grid 4

Fig. 5. Illustration of data dissemination process

- **Data Dissemination Process**

After the center collects data from four grids, one new packet with the information of the whole network is produced. More energy is consumed if we disseminate the new packet, because every node receives redundant data except the center node. We use an efficient method to deal with the problem. Center sends the right packet to the corresponding gateway neighbor, shown in Fig.5. Data from other grids will be aggregated and sent to grid 1. Relaying node will block the data which has been sent by itself. Hence, the data coming into grid 1 is reduced for avoiding redundancy. This packet is transmitted from gateway to root with sensing data of the relaying nodes combined. After root receives the packet, it gathers all the data of the whole sensor network. The root node performs the similar transmission as the center node to the children. The $data(F,G)$ will be blocked when root T send the information of the whole network to node G. Each downstream node will insert local data into the received packet, until transmission terminates at the leaf node. In the end, every node in the network receives data of all the other nodes.

4 Simulations

In this section, we evaluate the performance of the proposed DEIST, in terms of network lifetime, energy consumption and delay, with the A1 and A2 Chains [1] using the NS-2 simulator. Sensor networks are randomly generated depending on predefined node number. We assume that the sensor nodes are distributed in a $200m \times 200m$ region. The initial energy assigned to each sensor node is identical, about 1J.

We are interested in the lifetime of the system and the average energy consumption of each round. Network lifetime is defined as the number of rounds until the first sensor dies in the system. Besides, delay is another important performance measure of sensor protocols. It can be measured as the time delay between the data packets received at the nodes and the data generated at the source nodes. We use the number of hops spent on transmission to describe the delay of the network. For each simulation, we create more than 100 random networks and average the results. The length of messages transported in the sensor network is assumed to be 2000-*bit* due to data aggregation performed in each node.

We first evaluate the energy performance of our DEIST against the other algorithms A1 and A2 chain. Fig. 6 clearly shows that DEIST significantly outperforms others in terms of network lifetime. In dense networks, our DEIST can prolong the lifetime at least 69% of chain protocols. The ending node of chain protocols is too far away from center node. Therefore, center node of chain protocols consumes more energy per round. Our DEIST has many candidate gateway nodes in each grid, so the energy consumed by center is less than that of A1 and A2 chains.

The average energy consumed per round is depicted in Fig.7. The figure shows that the average energy cost decreases slowly for the three algorithms. That is because the average distance between two nodes is decreased with the number of sensor nodes increased. But our DEIST wins again.

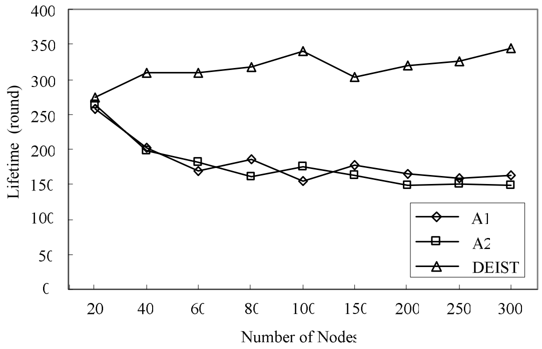


Fig. 6. Lifetime versus number of nodes with 2000-bit packets, in a 200m ×200m area

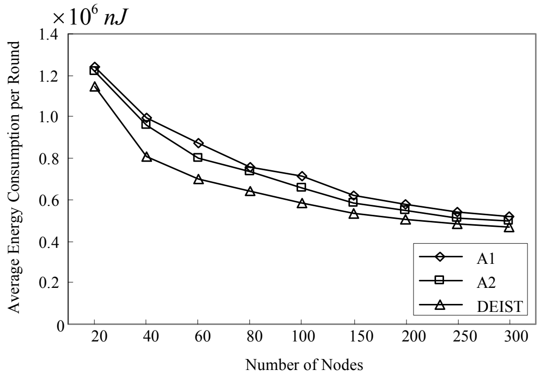


Fig. 7. Average energy consumption per round versus number of nodes with 2000-bit packets, in a 200m ×200m area

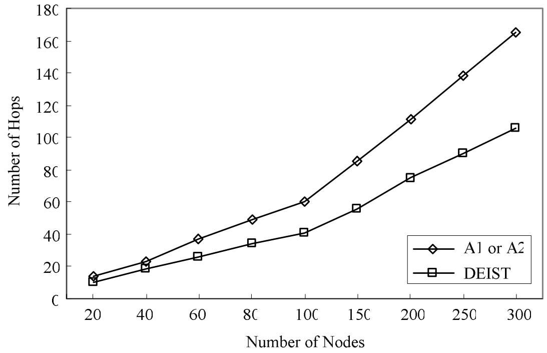


Fig. 8. Delay versus number of nodes with 2000-bit packets, in a 200m ×200m area

At last, we study the delay performance of the three algorithms. The results from the second set of simulation with delay are presented in Fig. 8, where the number of nodes is varied between 20 and 300.Obviously, Chain protocols have the same delay according to our assumption. With the help of root node, the performance of DEIST algorithm improves at least 35% on that of A1 or A2 chain.

5 Conclusion

In this paper we addressed the issue of data exchange in wireless sensor networks. We proposed a new protocol, called DEIST, and provided an analysis of the approach in comparison to the other protocols. Specifically, we use a novel algorithm to construct local IST in the four divided grids respectively. Simulation results showed that our DEIST protocol is not only able to effectively prolong the lifetime but also efficiently reduce the delay of the sensor networks. In the future, we will study the bottleneck problem of the local data center and the global data center. Besides, we will consider efficient data exchange algorithm for mobile sensor networks.

References

1. Du, K., Wu, J., Zhou, D.: Chain-Based Protocols for Data Broadcasting and Gathering in Sensor Networks. In: IPDPS 2003. International Parallel and Distributed Processing Symposium, p. 260a (2003)
2. Kim, S., Son, S.H., Stankovic, J.A., Li, S., Choi, Y.: SAFE: A Data Dissemination Protocol for Periodic Updates in Sensor Networks. In: ICDCSW 2003. 23rd International Conference on Distributed Computing Systems Workshops, p. 228 (2003)
3. Lindsey, S., Raghavendra, C.S.: PEGASIS: Power Efficient Gathering in Sensor Information Systems. In: Proceedings of the IEEE Aerospace Conference (March 2002)
4. Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H.: Energy-Efficient Communication Protocol for Wireless Microsensor Networks. In: HICSS. 33rd Hawaii International Conference on System Sciences, vol. 8, p. 8020 (2000)
5. Rajagopalan, R., Varshney, P.K.: Data-aggregation techniques in sensor networks: A survey. IEEE Communications Surveys & Tutorials 8(4), 48–63
6. Katz, R.H., Kahn, J.M., Pister, K.S.J.: Mobile Networking for Smart Dust. In: Proceedings of 5th ACM/IEEE Mobicom Conference (1999)
7. Sabbineni, H., Chakrabarty, K.: Location-Aided Flooding: An Energy-Efficient Data Dissemination Protocol for Wireless Sensor Networks. IEEE Transactions on Computers 54(1), 36–46 (2005)
8. Kahn, J.M., Katz, R.H., Pister, K.S.J.: Next century challenges: mobile networking for "Smart Dust". In: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking, Seattle, Washington, United States, pp. 271–278 (August 15-19, 1999)
9. Min, R., Bhardwaj, M., Cho, S.-H., Shih, E., Sinha, A., Wang, A., Chandrakasan, A.: Low-Power Wireless Sensor Networks. In: VLSID 2001. The 14th International Conference on VLSI Design, p. 205 (2001)
10. Intanagonwiwat, C., Govindan, R., Estrin, D.: Directed diffusion: a scalable and robust communication paradigm for sensor networks. In: Proceedings of the 6th annual international conference on Mobile computing and networking, Boston, Massachusetts, United States, pp. 56–67 (August 06-11, 2000)
11. Krishnamachari, B., Estrin, D., Wicker, S.: Modelling Data-Centric Routing in Wireless Sensor Networks. In IEEE INFOCOM http://citeseer.ist.psu.edu/554273.html

Low-Latency Mobile IP Handover Based on Active-Scan Link Layer Assisted FMIPv6

Chun Hsia and Chunhung Richard Lin

Department of Computer Science and Engineering, National Sun Yat-Sen University, Kaohsiung 804, ROC

Abstract. IEEE 802.11-based wireless local area networks (WLANs) have been set up in many public places in recent years. It provides convenient network connectivity to mobile nodes (MNs) and allows users moving from one wireless network to another. With mobility protocol support, such as Mobile IPv6 (MIPv6), people can roam across wireless IP subnets without loss of network-layer connectivity. However, the handover latency may make users uncomfortable in MIPv6. To support seamless handover, an enhanced MIPv6 scheme, Fast Handovers for Mobile IPv6 (FMIPv6) [1], was been proposed. In order to further reduce the handover latency, integration IEEE 802.11 and MIPv6 is necessary. Unfortunately, when integrating the IEEE 802.11-based standard with FMIPv6, FMIPv6 always fails to perform predictive handover procedure and results in reactive handover. It is because of the protocol nature of IEEE 802.11 and the weak relationship between IEEE 802.11 and FMIPv6. Furthermore, a MN can not receive packets destined to it as it sends the FBU to the original access router (OAR). This would cause unnecessary packet loss and make the predictive handover have more packet loss then reactive. Those issues will cause quality of services degradation and make real-time application unreachable. In this paper, a low-latency MIPv6 handover scheme will be proposed. It is a FMIPv6-based scheme, which is based on an active-scan scheme link layer assistance. It has the advantage of FMIPv6 and can reduce the unnecessary packet loss when the handover occurs. Also, with the active scheme assistance, it can avoid the longest phase that IEEE 802.11 will enter, and can lower the handover latency.

1 Introduction

Because of the demand for mobility, it is important to provide continuously connection when a MN moves, especially when the MN changes its attachment point. A MN performs the layer2 (L2) and layer3 (L3) handover procedure when it attaches to a new access point (AP) and changes its attachment point to the new access router (NAR) respectively. During the handover, a MN is unable to send or receive packets. If the delay or the latency is large, the ongoing session will be disrupted and the real-time applications will be unreachable. To reduce the handover latency caused by MIPv6, an enhanced MIPv6 draft, Fast Handover for Mobile IPv6, has been proposed. It has predictive handover and reactive handover. The predictive handover always relies on a L2 trigger and initializes the L3 handover procedure before the L2

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 165–173, 2007.

handover completes, while the reactive handover always starts the handover procedure after the L2 handover has completed. It is observation that predictive handover should cause lower packet loss because it builds a tunnel between the OAR and the NAR in advance. Unfortunately, in fact, the predictive has more packet loss than the reactive. It is because the protocol nature of FMIPv6. When a MN starts predictive handover procedure, it can not receive packets from current link. This may cause upper-layer handover procedure wait for its completion. In order to reduce the whole handover latency, the L2 handover procedure must be integrated into upper-layer schemes.

In this paper, we propose a low-latency mobile IP handover scheme, which is based on an active-scan link layer assisted FMIPv6. It monitors channels continuously, tries to find a better channel actively and notifies FMIPv6 to build a tunnel for the oncoming handover in advance. Besides, the OAR still forwards packets to the MN, instead of forwarding packet to the NAR in the original FMIPv6 when the predictive handover occurs. This scheme not only can reduce the latency cause by handover but also can reduce the packet loss caused by FMIPv6.

The rest of this paper is organized as follows. Section 2 summarizes some related work and makes some description of them. A detailed description of the proposed scheme is described in section 3. In section 4, a simple experiment has been made and the measures are discussed. Finally, we conclude this paper in section 5.

2 Related Works

With the growth of real-time applications, the latency and packet loss caused by mobility become an important issue in Mobile Networks. The most discussed topics are to reduce the IEEE 802.11 link-layer handover latency and to reduce the MIPv6 handover latency because of their popularity.

The IEEE 802.11 link-layer handover procedure can be split into three sequential phases: *potential*, *scan* and *auth*. The delay of the *scan* phase costs lots of time and varies from different cards. [2] shows the time-consumption during the handover between different APs and different MNs. Besides, [3] gives the measuring results of all three phases on different cards and proposes that tuning two parameters, *MinChannelTIme* and *MaxChannelTime*, to optimize the handover latency.

To reduce the latency that the *scan* phase causes, a scheme called Selective Channel Probing had been proposed. [4] and [5] are the related works of this scheme. Besides, [4] uses a caching scheme to store a list of MAC addresses of APs adjacent to current one and further improves the handover procedure. However, [5] relies on the IAPP to calculate the neighbor graph and selectively scans the APs in the neighborhood.

Mobile IP (MIP[6], MIPv4[6] and MIPv6[7]), the most popular Mobility Protocol, provides IP level mobility and allows MNs roaming across wireless networks without loss of network-layer connectivity. Unfortunately, the latency caused by MIP is longer. Early research focused on home registration latency which is called triangular problem. For example, Cellular IP (CIP) [8] is a technique to use the proprietary control messages for the location management. The messages will be routed in a regional area therefore speeding up the registrations and reducing the handover latency. Hierarchical MIP (HMIP) [9] is an extension of MIP, it employs a hierarchy of the foreign agents (FAs) to locally handle MIP registrations. Registration messages establish tunnels between the neighboring FAs along the path from the MN to a gateway FA. In addition, some research that focuses on other problems has been proposed. A method proposed by Yokota et. al. [10] named LLAMIP uses an AP and a dedicated MAC bridge to reduce the transmission interruptions in both the forward and reverse directions. FMIPv6, a new MIP extension, was proposed to address how a MN can send packets as soon as it detects a new subnet link and how to deliver packets to the MN as soon as it is detected by the NAR. [11] named S-MIP builds an architecture on top of the HMIPv6 and FMIPv6, and uses the location tracking and movement pattern detection to continuously track the MN's location and the next movement direction. By using those methods, it achieves the seamless handover.

Unfortunately, these MIP-based approaches always involve the L2 handover latency in the whole handover latency. To reduce the whole handover latency, the L2 and L3 handover procedure must be integrated together. Hence, another improvement which combines the L2 scheme and L3 protocol to speed up the handover procedure had been proposed. In LMIP [12], Sharma et.al. used the information from NIC driver to speed up the movement detection. Also, they designed a MIP advertisement caching and relay proxy to reduce the proxy advertisement or solicitation delay, and further reduce the L3 handover latency. [13] proposes an IAPP-based protocol to achieve zero-delay latency of the movement detection. It uses IEEE 802.11 re-association request message to see if the L3 handover is needed. [14] employs a Location Association Server (LAS) which maintains the location information, handover-to relationships, and AP/MA or AP/DHCP associations for a set of APs to help the MN for being aware of the nFA's location. Then, the MN can speed up the L3 handover procedure by pre-registration. Furthermore, [15] compared the predictive and reactive handover procedure in FMIPv6. It found that the reactive handover has shorter handover latency than predictive. This is because a MN can not receive any packet destined to it when it has sent the FBU messages. Even if it is still at the OAR's link.

In this paper, we propose a low-latency mobile IP handover scheme, which is based on an active-scan link layer assisted FMIPv6. It is a FMIPv6-based scheme with a active-scan link layer assistance. It makes FMIPv6 can get benefit from predictive handover. Also, it proposes a scheme to reduce unnecessary packet loss caused by FMIPv6.

3 Low-Latency Mobile IP Handover Based on Active-Scan Link Layer Assisted FMIPv6

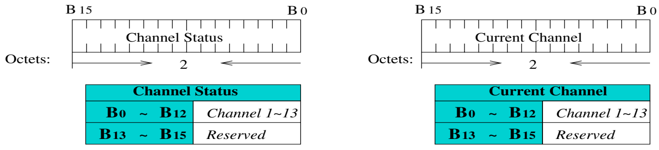
In this section, a scheme named low-latency mobile IP handover based on active-scan link layer assisted FMIPv6 is proposed. The active-scan scheme can monitor the channel status in the background, decides whether the handover is needed and is able to promptly trigger upper layer protocol to initialize the handover procedure. It helps FMIPv6 to initialize the handover promptly and avoids unnecessary latency. Also, a modification of FMIPv6 should be proposed. This can solve the early-start problem in FMIPv6 and can let FMIPv6 get benefit from predictive handover.

A. Active-Scan Link Layer Assistance

The active-scan link layer aims to bypass the most time-consumption phases in the L2 handover procedure and provides prompt information for upper layer. Hence, it monitors the channels it may operate in to bypass the *potential* phase, and triggers the upper layer as soon as possible while the handover is needed. Nevertheless, monitoring the channels may cause unnecessary packet loss or delay the packet transmission. To avoid the packet loss and unnecessary delay, the monitor procedure must be low enough. To this end, a Selective Channel Probing scheme should be used. This proposal also provides a Selective Channel Probing scheme. In addition, the proposed scheme can actively probe the channels without waiting for the *potential* phase detecting the needed handover. It can reduce the L2 handover delay intelligently. The details of the scheme are described below.

A.1. Selective Channel Probing

Firstly, two 2-byte fields, *CURRENT_CHANNEL* and *CHANNEL_STATUS*, are added into the probe request and response message respectively. They are used to avoid all channels being probed in the scan phase, as shown in Fig. 1.



Channel Status	
B0 ~ B12	*Channel 1~13*
B13 ~ B15	*Reserved*

Current Channel	
B0 ~ B12	*Channel 1~13*
B13 ~ B15	*Reserved*

(a) for probe response message (or beacon)

(b) for probe request message

Fig. 1. New elements in IEEE 802.11 messages

Also, a parameter, *CHANNEL_LIST*, in the AP and MN are utilized to record which channel should be probed in the scan phase. The details are described as follows.

CHANNEL_STATUS records the channels that MNs should probe. It is derived from *CHANNEL_LIST*. B0 to B12 represent the used status of channel 1 to 13 respectively, and B13 to B15 are reserved. *CURRENT_CHANNEL* records the channel where the MN now operates. Similarly, B0 to B12 represent the current operated channel. B13 to B15 are reserved, too. For example, if a MN uses channel n to make communications, it will set Bn-1 to 1 and keep other subfields to 0 into *CURRENT_CHANNEL*. Then, it will send the probing request message to an AP in the scan time. After receiving the message, the AP will refresh the *CHANNEL_LIST* by *CURRENT_CHANNEL* field. A MN, when receiving a beacon (or a probe response message from current operating channel), has to record the *CHANNEL_STATUS* into its *CHANNEL_LIST*. Then, it will only probe the channels recorded in the *CHANNEL_LIST* at next probing time.

We only provide a simple Selective Channel Probing scheme there, and assume that there are enough MNs in the overlap area to report their operating channels to their APs. It is used to simplify our simulation. For different consideration, other Selective Channel Probing schemes can be used to substitute our scheme directly.

A.2. Active-Scan Link Layer

The *potential* phase, however, is the longest phase in the L2 handover procedure and widely varies among different cards as [3] reported. This is because IEEE 802.11 only specifies the mechanisms to implement the handover but how to realize is not described. In order to reduce the FMIPv6 handover latency, the MN must be aware of the bad link status, get the necessary information and initialize the predictive handover early. In this subsection, we provide an active-scan link layer scheme which will monitor the channel status, find the better channels and further bypass the *potential* phase.

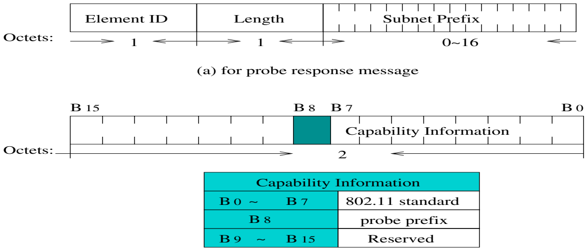
Instead of entering the *scan* phase after the *potential* phase, this active-scan link layer probes the channels in advance. Furthermore, it continuously probes the channels every *PROBE_INTERVAL*. The *PROBE_INTERVAL* is a counter, which subtracts 1 every beacon time. When it reaches to 0, this active scheme will enter into the *scan* phase which tries to find a better channel and decides whether the handover is needed. After probing the channels, the *PROBE_INTERVAL* will be reloaded for the next *scan* phase.

B. Enhanced FMIPv6

Although the Active-Scan Link Layer Assisted method can reduce the handover latency that FMIPv6 may experience, FMIPv6 still can not get benefit from predictive handover procedure. This is because FMIPv6 can't complete all the necessary works in the OAR's link. With this active method support, however, a FMIPv6 MN can initialize the L3 handover early. Unfortunately, the early start FMIPv6 will cause more packet loss. It is the nature of the FMPv6 protocol, because the OAR will stop forwarding packets to the MN when it receives the handover message (the FBU message) in its link. To integrate the L2 scheme into FMIPv6 and reduce the whole handover latency, a MN must be able to complete the necessary works as soon as possible and must be modified to reduce the unnecessary packet loss that the early start FMIPv6 may cause.

B.1. A L2 Assisted Method for FMIPv6

In FMIPv6, a MN must discover available APs and then request the subnet information corresponding to the discovered APs. However, discovering available APs and request the subnet may cost lots of time. They may make the handover longer and cause the packet loss. In this subsection, the probing messages are utilized to speed the FMIPv6 handover procedure, as shown in Fig. 2.



(a) for probe response message

Capability Information	
B 0 ~ B 7	802.11 standard
B 8	probe prefix
B 9 ~ B 15	Reserved

(b) for probe request message

Fig. 2. New option for probing the subnet information

Firstly, we add an *Information Element* into the probe response message and add an option into the *Capability Information* field in the request message. The *Element ID* in Fig. 2(a) is a reserved number in IEEE 802.11. The *Length* is a variable which is range from 0 to 128, and means the number of bits in the subnet prefix. The subnet prefix is the IPv6 prefix and range from 0 to 128 bits. For the request message, we add an option in the *Capability Information* field. B8 to B15 are reserved and the B8 is used to query the network prefix, shown as Fig. 2(b).

When an AP receives the router advertisement message, it records the IPv6 subnet prefix. A MN will request the prefix when it wants to initialize the FMIPv6 handover procedure. It sets B8 in the *Capability Information* field and sends the probe request message to the AP. The AP will response the prefix to the MN. As receiving the response, the MN records the prefix and uses the prefix for the oncoming handover. Because the active method probes the channel periodically, the prefix can be refreshed. With the active method support, the MN can not only reduce the L2 handover delay, but also get the necessary information in advance. Then, it can help the FMIPv6 handover procedure start promptly.

B.2. The Problem of Early Start FMIPv6

As we described above, an OAR will stop forwarding packets to the MN when it has received the FBU message in FMIPv6. However, when the MN is still in the OAR's link, it can't receive any packet destined to it. This may cause unnecessary packet loss. This subsection proposes a method which tries to utilize two FBU messages to solve the problem.

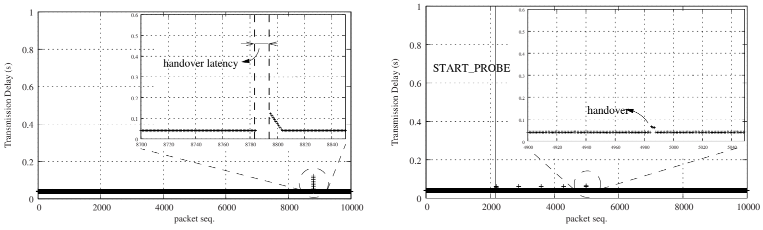
When the handover is needed, the MN follows the FMIPv6 standard which sends the FBU message to notify the OAR of the oncoming handover. After receiving the FBU from its link, the OAR builds a tunnel for the oncoming handover like the original FMIPv6 does, but it keeps forwarding packets to the MN. When a MN decides to switch to the new AP, it will send the second FBU message to notify the OAR of its leaving. Then, the OAR will forward the packet to the NAR as the FMIPv6 standard operates. The first FBU is used to complete all the works that predicative handover should finish at the OAR's link except binding current CoA to the new CoA. The second FBU, however, is used to notify the OAR of binding current CoA to the new CoA and then the OAR can forward packet to the NAR.

4 Simulation and Analysis

In this section, the simulations of our proposal are presented and compared with IEEE 802.11 and FMIPv6. There were three wireless networks in the experimental networks. Two of them, AP_1 and AP_2, were belong to the same IPv6 subnet and the other, AP_3, is in a different network segment. The simulations were performed by omnet++ with IPv6Suit. The transmission radius of AP is 250m, which is for an outdoor environment. The overlap length is 100m. The wireless link speed is based on IEEE 802.11b and the open system was used to be the default authentication algorithm. We only focus on walking speed which varies from 1m/s to 5m/s.

A. L2 Handover Latency

To evaluate the performance improvement of the active scheme, a MN moved from the communication range of AP_1 to AP_2. The CN was used to transmit 64-byte UDP packets to the MN every 10 ms. 10000 packets was sent in each experiment. The Selective Channel Probing was used in IEEE 802.11 standard and the active scheme. The results are shown in Fig. 3. The transmission delay means the duration from a packet sent out to the packet received.



(a) MN with IEEE 802.11 standard (b) MN with the Active Scheme

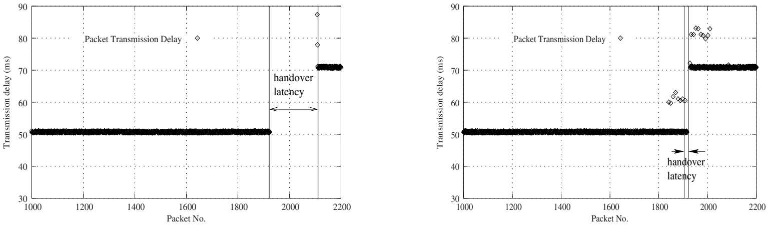
Fig. 3. L2 Handover Latency Experiment

A shown in Fig. 3(a), there is packet loss since the MN can not receive any packet during the link-layer handover. Moreover, after the handover, the transmission delay is longer than before. It is because that the packets may be buffered at the AP and delay their transmission time. Obviously, the latency of IEEE 802.11 can not meet the requirements of real time applications. Even the selective channel probing is used.

In Fig. 3(b), because the MN with the active scheme can keep probing the channels in range, there is no packet loss but there are some packets with higher transmission delay. However, the number of packets with higher transmission time is small and the transmission delay is also little longer than the usual packets. They will not make the significant effect on most applications. The improvement of the active scheme can be observed clearly in this experiment.

B. FMIPv6 Handover Experiment

Although the active scheme can reduce the L2 handover latency, the whole handover latency must be evaluated, too. For this purpose, the MN migrated from the



(a) **FMIPv6 integrated with IEEE 802.11** (b) **Enhanced FMIPv6 with the active scheme**

Fig. 4. The whole handover latency

communication range of AR_1 (AP_2) to AR_2 (AP_3). The CN was used to transmit 64-byte UDP packets to the MN at every 10 ms intervals. 4000 packets were sent in each experiment. The results are shown as Fig. 4.

In Fig. 4(a), FMIPv6 integrated with IEEE 802.11 is measured. It has more packet loss when the handover occurs. This is because FMIPv6 would suffer the IEEE 802.11 handover latency and results in reactive handover. Even if IEEE 802.11 can trigger FMIPv6 to initialize the predictive handover, it will have more packet loss. This is discussed in [15]. Moreover, when L2 handover procedure is proceeding, the upper layer can't send or receive packets. Another observation is that the transmission delay is higher than before after the handover. This is because we didn't use the route optimization. The packets will arrive at the HA first and then be forwarded to the MN. In Fig. 4(b), the enhanced FMIPv6 with the active scheme is discussed. The most interesting thing is that the handover latency is much lower. Because the active scheme can help the FMIPv6 to do the best handover decision, the enhanced FMIPv6 avoids the longest phase in the L2 handover procedure. Also, after sending the FBU message, a MN can still receive the packets at the OAR's link. This helps FMIPv6 to lower the unnecessary packet loss. However, around the handover, there are some packets with high transmission delay. It is caused by the usual probing routine. The active scheme keeps probing the potential channels when the channel quality is bad. This makes the packets have higher transmission delay.

5 Conclusions

This paper proposes an integration handover scheme which is based on an active link layer and FMIPv6. The active scheme continuously monitors the signal qualities of the potential channels in its communication range and can decide the most suitable handover opportunity. It can not only reduce the L2 handover latency but also help FMIPv6 to initialize the handover procedure promptly. Although it may cause some packets with high transmission delay, it avoids the most time-consumption phase in IEEE 802.11. Also, this scheme makes FMIPv6 have enough time to complete the works that the predictive handover procedure should finish at the OAR's link. With this active scheme support, FMIPv6 can not only lower the handover latency but also reduce the packet loss caused by the handover. Besides, the original FMIPv6 is also modifies to reduce the unnecessary packet loss. It makes the OAR still forward packets to the MN when it firstly receives the FBU and start forwarding the new coming packets to NAR when receiving the FBU again. Because the active scheme can achieve lower handover latency and make FMIPv6 finish the necessary work at the OAR's link, it lower the whole handover latency. Moreover, because of the lower handover latency, the real-time applications become more possible.

Reference

[1] Koodli, R. (ed.): Fast Handovers for Mobile IPv6, RFC 4068 (July 2005)

[2] Mishra, A., Shin, M., Arbaugh, W.: An Empirical Analysis of the IEEE 802.11 MAC Layer Handoff Process. ACM SIGCOMM Computer Communication 33(2) (April 2003)

[3] Wang, G.-Y.M., Lin, C.R.: An Optimization Scheme for the IEEE 802.11 Link-Layer Handoff Process. Journal of Information Science and Engineering (accepted)

[4] Shin, S., Forte, A.G., Rawat, A.S., Schulzrinne, H.: Reducing MAC Layer Handoff Latency in IEEE 802.11 Wireless LANs. In: ACM MobiWac 2004 (October 2004)

[5] Kim, H.S., Park, S.H., Park, C.S., Kim, J.W., Ko, S.J.: Selective Channel Scanning for Fast Handoff in Wireless LAN using Neighbor Graph. In: Niemegeers, I.G.M.M., de Groot, S.H. (eds.) PWC 2004. LNCS, vol. 3260, Springer, Heidelberg (2004)

[6] Perkins, C.: IP Mobility Support for IPv4. RFC3344, IETF (August 2002)

[7] Johnson, D., Perkins, C., Arkko, J.: Mobility Support in IPv6. RFC3775, IETF (June 2004)

[8] Valkó, A.G.: Cellular IP: A New Approach to Internet Host Mobility. ACM SIGCOMM Computer and Communication Review 29(1), 50–65 (1999)

[9] Gustafasson, E., Jonsson, A., Perkins, C.: Mobile IP regional registration. Internet draft, draft-ietf-mobileip-reg-tunnel-09.txt (June 2004)

[10] Yokota, H., Idoue, A., Hasegawa, T., Kato, T.: Link Layer Assisted Mobile IP Fast Handoff Method over Wireless LAN Networks. In: MOBICOM 2002 (2002)

[11] Hsieh, R., Zhou, Z.G., Seneviratne, A.: S-MIP: A Seamless Handoff Architecture for Mobile IP. In: Proceedings of IEEE INFOCOM (March 2003)

[12] Sharma, S., Zhu, N., Chiueh, T.: Low-Latency Mobile IP for Infrastructure- Mode Wireless LANs. IEEE Journal on Selected Areas in Communication 22(4), 643–652

[13] Samprakou, I., Bouras, C.J., Karoubalis, T.: Fast and Efficient IP Handover in IEEE 802.11 Wireless LANs. In: ICWN 2004, pp. 249–255 (June 2004)

[14] Tseng, C.C., Yen, L.H., Chang, H.H., Hsu, K.C.: Topology-Aided Cross-Layer Fast Handoff Designs for IEEE 802.11/Mobile IP Environments, IEEE Communications Magazine (December 2005)

[15] Dimopoulou, L., Leoleis, G., Venieris, I.S.: Analysis and Evaluation of Layer 2 Assisted Fast Mobile IPv6 Handovers in a WLAN Environment. In: ISCC 2005. Proceedings of the 10th IEEE Symposium on Computers and Communications (2005)

Achieving Efficiency Channel Utilization and Weighted Fairness in IEEE 802.11 WLANs with a P-Persistent Enhanced DCF

Jain-Shing Liu[1] and Chun-Hung Richard Lin[2]

[1] Department of Computer Science and Information Management, Providence University, Taiwan, R.O.C
[2] Department of Computer Science and Engineering, National Sun Yat-Sen University, Taiwan, R.O.C
{alfred.hofmann,ursula.barth,ingrid.beyer,christine.guenther,
frank.holzwarth,anna.kramer,erika.siebert-cole,lncs}@springer.com
http://www.springer.com/lncs

Abstract. Fair allocation of bandwidth and maximization of channel utilization are two important issues when designing a contention-based wireless medium access control (MAC) protocol. However, fulfilling both design goals at the same time is very difficult. Considering the problem in the IEEE 802.11 wireless local area networks (WLANs), in this work we propose a method using a p-persistent enhanced DCF, called *P-IEEE 802.11 DCF*, to achieve the weighted fairness among multiple priority classes in a WLAN. The key idea of this method is that when the backoff timer of a node reaches zero, the transmission probability is properly controlled to reflect the relative weights among data traffic flows so as to maximize the aggregate throughput and to minimize the frame delay at the same time. In particular, we obtain the optimal transmission probability based on a theoretical analysis, and also provide an approximation to this probability. The derived optimal and approximation are all evaluated numerically and simulated with different scenarios. The results show that the proposed method can fulfill our design goals under different numbers of priority classes and different numbers of hosts.

1 Introduction

1.1 Motivation and Problem Statement

IEEE 802.11 [1] has become one of the most important and successful MAC protocols for wireless infrastructure and infrastructureless (*ad hoc*) LANs. It simultaneously employs a mandatory contention-based channel access function called Distributed Coordination Function (DCF), and an optional centrally controlled channel access function called Point Coordination Function (PCF). However, the popularity of IEEE 802.11 market is mainly due to DCF, while PCF is barely implemented in the current products due to its complexity and inefficiency for the normal data transmission. In contrast, DCF adopts a carrier sense multiple

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 174–184, 2007.

access with collision avoidance (CSMA/CA) with binary exponential back-off, which enables fast installation with minimal management and maintenance costs in the networks. With the aid of DCF, IEEE 802.11 WLAN becomes the most successful consumer wireless platform, just as its wired counterpart, Ethernet. Now, with the rapid world-wide deployment of WLANs, the corresponding devices, such as wireless interface cards for notebook and desktop PCs, turn out to be the most popular consumer electronics in the world.

Given its popularity, IEEE 802.11 DCF, however, does not provide service differentiation, such as throughput and delay guarantees, among connections of different priorities. Moreover, it also does not consider fair allocation of bandwidth in the networks. In fact, the above two missed in DCF, have been identified as the most important issues when designing the wireless card of next generation to provide Qualify of Service (QoS) in WLANs. Unfortunately, the two deign goals conflict with each other in usual. It can be easily identified in the wireless networks that the maximum channel utilization could be achieved if only one node is allowed to transmit continuously without back-off, while all the other nodes are starved. The maximum throughput achieved by the monopolistic behavior obviously conflicts with the goal of fair share.

In this paper, we study the challenging problem to achieve both goals simultaneously, especially for data communications in WLANs. To this end, the ideal weighted fairness should be defined at first. Assume that there are N different priority classes. Each class i is characterized by a positive weight, ψ_i, with the assumption of $1 = \psi_1 > \psi_2, ..., > \psi_n > 0$. Assume further that each node carries only one traffic flow, with f_i denoting the set of nodes carrying class i traffic. Let $w_i^k(t_b, t_e)$ be the amount of class i traffic transmitted by node $k \in f_i$ during the time interval $[t_b, t_e]$. In order to achieve fair share to all traffic flow, it requires

$$\frac{w_i^k(t_b, t_e)}{\psi_i} = \frac{w_j^{k'}(t_b, t_e)}{\psi_j},$$
$$\forall i, j \in \{1, ..., N\}, \ \forall k \in f_i, \ \forall k' \in f_j \tag{1}$$

As shown above, the ideal weighted fairness cannot be actually achieved since data transmitted on a real network is packetized. However, when considered with IEEE 802.11 WLANs, each data packet in the higher layer is fragmented into smaller MAC frames for transmission, which provides a reasonable assumption that each data flow has the same MAC frame size. Let $P_{s,i}$ be the probability that a MAC frame is transmitted from a node in class i and successful. With this, it is considered that all the traffic flows within a WLAN would fairly share the wireless medium and the weighted fairness in the WLAN is achieved, in a probabilistic sense, if the following condition holds

$$\frac{P_{s,i}}{\psi_i} = \frac{P_{s,j}}{\psi_j}, \ \forall i, j \in \{1, ..., N\} \tag{2}$$

Given the above, the objective of this work is then to propose a method that can 1) achieve the weighted fairness among different priority data flows while maximizing the aggregate throughput and minimizing the frame delay, and 2) remain in compliance with the legacy 802.11 DCF without requirement of changes in its existing frame formats and access procedures.

1.2 Related Works

Since IEEE 802.11 becomes the de facto standard for WLANs, there are quite a lot of related works proposed either for obtaining its theoretical limits or for improving its performances. Specifically, many related works have been done to develop scheduling algorithms for wireless networks to achieve weighted fairness. However, most of them are centralized or polling-based protocols. Recently, with the distributed EDCF in IEEE 802.11e, some works have also been done for service differentiation by using different priority schemes based on, for example, setting different IFS, CW, or back-off parameters specified in the MAC. However, these previous works are usually conducted for either performance maximization or weighted fairness in the WLANs.

Only few related works may take both design goals into account with a distributed mechanism. In these few works, [2] proposes a method to achieve weighted fairness in IEEE 802.11e. However, it considers only 2 classes without performance maximization, and provides no general solution when the number of classes larger than 2. [3] attempts to deal with this problem in a multi-hop wireless network subject to a minimum fairness guarantee, which is different from the issues we address on the WLANs. [4] provides a so-called P-MAC that modifies the DCF to achieve the two goals, but it uses a constant contention window and requires modifications to the DCF. [5] extends the work in [6], and derives a value p for each class to maximize the system capacity while ensuring a user-specified utilization ratio. However, unlike ours, this work adopts another modeling scheme that determines the back-off interval with a p-persistent transmission probability sampling from a geometric distribution.

2 P-IEEE 802.11 DCF

To deal with the weighted fairness problem and fulfill the design objective mentioned previously, we choose to extend the capability of P-IEEE 802.11 DCF protocol in [7] that does comply with the legacy 802.11 DCF and require no changes in the existing frame formats and access procedures. As shown in Fig. 1, the P-IEEE protocol uses a separate layer between the standard access scheme and the physical layer to calculate the transmission opportunity for a node. When carried out, each node filters its transmission attempt based on the decision made in this layer. If the decision is positive, the frame under transmission is delivered by the normal DCF. Otherwise, the frame is deferred with a new back-off interval, just as that it encounters a collision in the legacy MAC.

Therefore, with this protocol a station can manipulatively control its transmission flow. However, it is still unknown that if a node in a certain priority class can

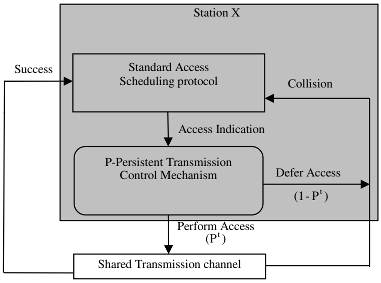


Fig. 1. The architecture of P-IEEE 802.11 DCF

decide the controlled p-persistent transmission probability for its transmission flow to cooperatively achieve the weighted fairness with other flows from different classes in the WLAN. For solving this problem, the controlled probability, called P_T, should be represented with a form that is analytically tractable in the multi-class environment. In this work, it is done with a simple non-uniform increasing function that can reasonably reflect the channel contention level sensed by a node in class i and back-off stage j. More precisely, with ϕ_i as the transmission factor for class i, the probability is represented by

$$P^t_{i,j} = 1 - \phi_i^{j+1} \tag{3}$$

3 Throughput of P-IEEE 802.11 Protocol in Multi-class Environment

For the throughput calculation in the multi-class environment, we first let the conditional collision probability of class i be P_i (as P in [8]). Given that, the successful transmission probability of a node can then be represented by $(1 - P_i) \cdot P^t_{i,j}$. With this probability, a node will reset its back-off timer to a value within $W_{i,0}$ (the minimum window size of class i). On the other hand, the failure transmission probability of a node can be given by $1 - (1 - P_i) \cdot P^t_{i,j}$. In this case, a node will defer its transmission to the next back-off stage, choosing a new back-off timer with a value within the window size of the stage. Besides, other non-null probabilities include the probability of 1 with which a back-off timer should decrease by 1 when the channel is sensed idle, and the probability of 1 with which a node should reset its contention window to $W_{i,0}$ when the maximum back-off stage m_i is encountered.

More precisely, for a node in calss $i \in [1, N]$, let $b(i, t)$ be the stochastic process representing the back-off timer $k \in [0, W_{i,j} - 1]$, and $s(i, t)$ be the process representing the back-off stage $j \in [0, m_i]$. Thus, at time t, the state of a node in class i can be modeled with a discrete-time Markov chain $\{b(i, t), s(i, t)\}$, and fully determined by $\{i, j, k\}$, as shown in Fig. 2. With the Markov chain, the non-null probabilities considered in above can be represented by

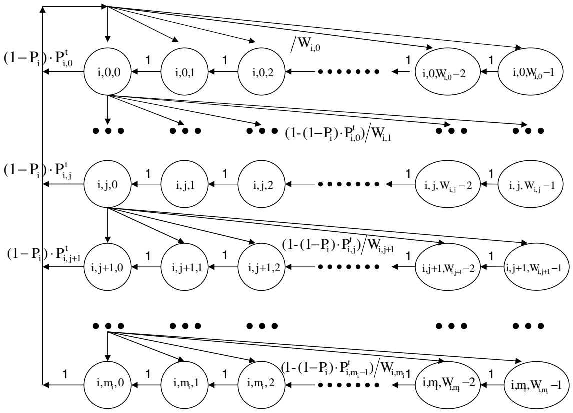


Fig. 2. The Markov chain model for P-IEEE 802.11 MAC with multiple classes

$$\begin{cases} P\{i,j,k|i,j,k+1\} = 1, \\ \qquad k \in (0, W_{i,j}-2),\ \ j \in (0, m_i) \\ P\{i,0,k|i,j,0\} \quad = \frac{(1-P_i)\cdot P^t_{i,j}}{W_{i,0}}, \\ \qquad k \in (0, W_{i,0}-1),\ \ j \in (0, m_i) \\ P\{i,j,k|i,j-1,0\} = \frac{1-(1-P_i)\cdot P^t_{i,j-1}}{W_{i,j}}, \\ \qquad k \in (0, W_{i,j}-1),\ \ j \in (1, m_i) \\ P\{i,0,k|i,m_i,0\} \quad = \frac{1}{W_{i,0}}, \\ \qquad k \in (0, W_{i,m_i}-1) \end{cases} \tag{4}$$

With some manipulations, we can lead the above to the stationary probability, $b_{i,j,k}$, for a node in class i with its back-off stage in j and back-off timer in k,

$$b_{i,j,k} = \frac{W_{i,j}-k}{W_{i,j}} \cdot \begin{cases} (1-P_i)\cdot \sum_{j=0}^{m_i-1} P^t_{i,j} \cdot \\ \prod_{k=0}^{j-1}(1-(1-P_i)P^t_{i,j})\cdot b_{i,0,0} + b_{i,m_i,0},\ j=0 \\ \left(1-(1-P_i)\cdot P^t_{i,j-1}\right)\cdot b_{i,j-1,0}, \quad 0 < j \leq m_i \end{cases} \tag{5}$$

Finally, the probability τ_i that a node transmits a frame in a randomly chosen time and the probability P_i that a station in the back-off stage senses the channel busy, both for class i, constitute a nonlinear system of equations as follow

$$\begin{cases} \tau_i \ = \sum_{j=0}^{m_i} P^t_{i,j} \cdot b_{i,j,0} \\ P_i = 1-(1-\tau_i)^{n_i-1} \cdot \prod_{h=1,h\neq i}^{N}(1-\tau_h)^{n_h} \end{cases} \tag{6}$$

where n_i denotes the number of nodes in class i. Not that because there are N classes, the system eventually has $2N$ unknowns τ_i and P_i, to be solved numerically.

Further, for the throughput calculation, we let P_{tr} be the probability of at least one transmission in a slot time. Similarly, let $P_{s,i}$ be the probability of a transmission that is successful for a node in class i (as defined previously), and

P_S be the probability that a successful transmission occurs in a slot time. In terms of τ_i, these probabilities can be represented by

$$P_{tr} = 1 - \prod_{h=1}^{N} (1 - \tau_h)^{n_h} \tag{7}$$

$$P_{s,i} = \tau_i \cdot (1 - \tau_i)^{n_i - 1} \cdot \prod_{h=1, h \neq i}^{N} (1 - \tau_h)^{n_h} \tag{8}$$

$$P_S = \sum_{i=1}^{N} n_i \cdot P_{s,i} = \sum_{i=1}^{N} \frac{n_i \cdot \tau_i}{1 - \tau_i} \cdot (1 - P_{tr}) \tag{9}$$

With these probabilities, we can express the throughput for a node in class i, S_i, and the overall system throughput, S, as the following ratios

$$S_i = \frac{P_{s,i} \cdot E[P]}{(1 - P_{tr}) \cdot \sigma + P_S \cdot T_s + (P_{tr} - P_S) \cdot T_c} \tag{10}$$

$$S = \frac{P_S \cdot E[P]}{(1 - P_{tr}) \cdot \sigma + P_S \cdot T_s + (P_{tr} - P_S) \cdot T_c} \tag{11}$$

where $E[P]$ denotes the average frame length, σ the duration of an empty slot time, and T_s and T_c the average times that the channel is sensed busy due to a successful transmission or a collision, respectively. For the values of these parameters, one may refer to [8].

4 Weighted Fairness with P-IEEE 802.11 DCF

In this section, we introduce a method to obtain weighted fairness among data traffic in different priority classes while maximizing aggregate throughput and minimizing frame delay. For doing so, we take $P_{s,i}$ in (8) into the weighted fairness in (2), and after some simple manipulations, we have

$$\begin{aligned} &\frac{\tau_i \cdot (1 - \tau_i)^{n_i - 1} \cdot (1 - \tau_j)^{n_j}}{\psi_i} = \\ &\frac{\tau_j \cdot (1 - \tau_j)^{n_j - 1} \cdot (1 - \tau_i)^{n_i}}{\psi_j}, \ \forall i, j \in \{1, ..., N\} \end{aligned} \tag{12}$$

Further, without loss of generality, we let $\tau_i = \tau_1$, which leads to

$$\begin{aligned} \tau_j &= \frac{\psi_j \cdot \tau_1 \cdot (1 - \tau_1)^{n_1 - 1}}{\psi_i \cdot (1 - \tau_1)^{n_1} + \psi_j \cdot \tau_1 \cdot (1 - \tau_1)^{n_1 - 1}} \\ &= \frac{\psi_j \cdot \tau_1}{\psi_1 - \psi_1 \cdot \tau_1 + \psi_j \cdot \tau_1} \end{aligned} \tag{13}$$

In other words, any $\tau_j, j \neq 1$ can be represented in terms of ψ_j, ψ_1, and τ_1. Thus, if the optimal transmission probability of class 1, τ_1^*, can be given, the optimal transmission probabilities of the other classes, τ_j^*s, can then be obtained with τ_1^* and the weights ψ_js. Given the above, we now go to the step to find τ_1^* that can achieve the maximum aggregate throughput and the minimum frame delay, which eventually leads to the optimal P_Ts as required. At first, we consider that for the maximum aggregate throughput as follows.

4.1 Optimal P-Persistent Probabilities of Transmission for the Maximum Aggregate Throughput

Optimal Method. For the maximum aggregate throughput in the multi-class environment, we need to know the relationship between τ_1 and the aggregate throughput in the system at first. To this end, we multiply (11) by $\frac{1}{P_S}$ to remove the probability P_S in the numerator part, leading to

$$S \overset{\times \frac{1}{P_S}}{=} \frac{E[P]}{T_s - T_c + \frac{P_{tr} \cdot T_c + (1-P_{tr}) \cdot \sigma}{P_S}} \tag{14}$$

From the above, we can see that if the denominator part can be minimized, the throughput S can be maximized. Thus, we define the following for optimization

$$S^f = \frac{P_S}{P_{tr} \cdot T_c + (1 - P_{tr}) \cdot \sigma} \tag{15}$$

It is clear from Eqs (7), (9) and (13) that S^f depends on τ_1. Thus we can solve the equation

$$\frac{d\,S^f}{d\,\tau_1} = 0 \tag{16}$$

to obtain the optimal probability τ_1^* that maximizes the aggregate throughput, and then apply (13) to obtain τ_j^*s for the other classes. Given that and the numbers of nodes in all priority classes, the optimal p-persistent transmission probabilities, $P_{i,j}^{t^*}$s, can be obtained with (3).

Approximation. However, as shown above, it is hard to find a closed-form solution for (16) when $N > 3$. Thus, we also consider an approximation to quickly calculate τ_i^* for each class i. The idea is obtained by observing the simplest case where there are only two nodes in the WLAN, and each node carries a traffic flow belonging to different classes. That is, $n_1 = 1$ and $n_2 = 1$. Further, letting $\psi_1 = 1$, S^f in (15) can thus be reduced to

$$S_{2(sim)}^f = \frac{\tau_1 \cdot (1 - \tau_2) + \tau_2 \cdot (1 - \tau_1)}{(1 - \tau_1) \cdot (1 - \tau_2) \cdot (\sigma - T_c) + T_c} \tag{17}$$

This simplified equation can be easily solved, which provides the optimal probability τ_1^* as

$$\tau_{1(sim)}^* = \frac{1}{1 + \sqrt{\frac{T_c}{\sigma} \cdot \psi_2}} \tag{18}$$

Based on this, we can make the following approximation for the multi-class case,

$$\tau_1^* \approx \frac{1}{n_1 + \sqrt{\frac{T_c}{\sigma} \cdot n_1 \cdot (\sum_{j=2}^{N} n_j \cdot \phi_j)}} \tag{19}$$

Given that, the other approximations of τ_j^*s can be obtained with (13). With these approximations, we can solve the nonlinear system equations in (6) for ϕ_i^*s. Finally, the approximated $P_{i,j}^{t^*}$s can be resulted from (3), with the ϕ_i^*s just obtained.

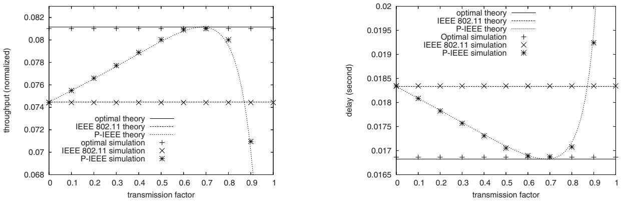
5 Performance Evaluation

In this section, we report on experiments made in order to verify the theoretical results derived previously. In the experiments, we implement P-IEEE with multiple classes on the Pythagor simulator [9] and let all nodes with IEEE 802.11a PHY be uniformly distributed in the WLAN. Each node has a flow with 2000-bytes of UDP packets toward a randomly chosen destination, resulting in the saturated throughput as required. With the assumption of no hidden terminal problem, P-IEEE and IEEE 802.11 MAC are both taken into account for the performance metrics under consideration, throughput, frame delay, and weighted fairness, with different scenarios. However, we show only the results of 6 Mbps data rate in IEEE 802.11a PHY, due to space limitations. Other results with different data rates have the same trend, and can be represented by that of 6 Mbps.

5.1 Single Class Case

Without loss of generality, we examine the performance model with a single class to verify the optimal P_T ($P_{i,j}^{t^*}$) derived. The optimal method derived from (14) is examined in the experimental environment with 10 nodes of the single class (class 1) uniformly distributed in the WLAN, and the transmission factor ϕ_1 changed from 0 to 0.9 with a step size of 0.1.

Figure 3 shows the results of theory and simulation with respect to the aggregate throughput and the average frame delay in (a) and (b), respectively. From



(a) (b)

Fig. 3. Experiment results of 10 nodes with a single class and 6 Mbps data rate: (a) the aggregate throughput, and (b) the average frame delay

this figure, we have several observations worth noting. First, the optimal value ϕ_1^* is about 0.66 for both throughput and delay. This is correctly predicted by our analysis showing that with either metric (throughput or delay), the optimal methods can lead to the same result. Second, when $\phi_1 = 0$ and thus $\phi_1^{j+1} = 0$, $\forall j$, P-IEEE has the same results as IEEE 802.11 MAC. It is expected because when the factor becomes zero, it means that $P_{1,j}^t = 1$, $\forall j$, and no p-persistent filtering is carried out. In the special case, P-IEEE exactly reverts to the IEEE 802.11 MAC that provides no filtering when its back-off counter reaches zero. Third, the simulation results are very close to the theoretical ones. This indicates that our model can provide very good accuracy of the performance evaluation.

5.2 Multiple Classes Case

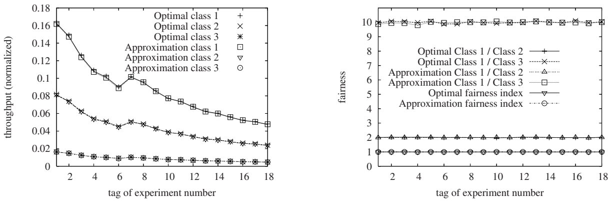
In the case study, we let the number of priority classes be 3, and in addition to the optimal methods, we also consider the approximation derived from (17). With these, we investigate the impact of the number of nodes in each priority class on the weighted fairness, and perform 18 different simulation experiments to verify the corresponding theoretical results. Each of these experiments is carried out using a different set of $(n_1; n_2; n_3)$, where n_i denotes the number of nodes in priority class i with $n_1 = 2, 5, 10$, $n_2 = 5, 10$ and $n_3 = 5, 10, 20$, and will be referred to as experiment 1 to 18. All these experiments are carried out with $\psi_1 = 1$, $\psi_2 = 0.5$ and $\psi_3 = 0.1$, which reasonably represents other possible setting of ψ.

In addition to the above, we conduct these experiments with two different fairness metrics to quantitatively evaluate these MACs. The first metric is the direct ratio between the performance metrics from these classes. More precisely, we consider the throughput ratio as $\frac{S_i}{S_j}$ (where S_i denotes the throughput of class i), and the delay ratio as $\frac{D_i}{D_j}$ (where D_i denotes the frame delay of class i). The second metric is the so-called *fairness index* in [4]. In this metric, S_f denotes the throughput of traffic flow f, and ψ_f denotes the associated weight. The throughput fairness index, F_s, is then defined as

$$F_s = \frac{\mu(S_f/\psi_f)}{\mu(S_f/\psi_f) + \alpha(S_f/\psi_f)} \tag{20}$$

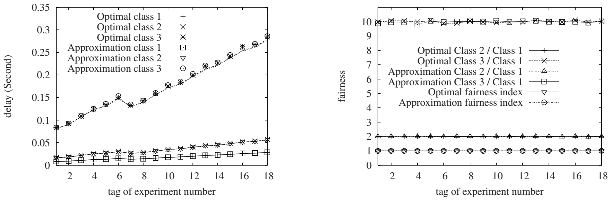
where μ and α denote, respectively, the mean and the standard deviation of S_f/ψ_f over all the active data traffic flows. Similarly, by replacing S_f with the delay of traffic flow f (i.e., D_f), the delay fairness index can be also obtained with equation (20).

Figure 4(a) shows the throughput results, with lines denoting the theory's and symbols denoting the simulation's. From this figure, we have three points of observation. First, the simulation results well match those of theory. This indicates that our analysis can correctly evaluate these methods. Second, the approximations have their throughputs very close to those of the optimal methods. The observation suggests that we can use these approximations to obtain nearly the same optimal throughputs with lower computational costs. Third, the



(a) (b)

Fig. 4. Throughput results of 3 classes: (a) throughput, and (b) the corresponding fairness index



(a) (b)

Fig. 5. Delay results of 3 classes: (a) delay, and (b) the corresponding fairness index

throughput of class 1 is almost 2 times (10 times) that of class 2 (class 3) in all scenarios. The fair share is expected and can be further confirmed in Fig. 4(b), in which the direct ratio between class 1 and 2 (class 1 and 3) is about 2 (10) and the fairness index is about 1 despite the scenarios. Both indicate the same results on the fairness. However the former represents the desired fair share between these two classes while the latter shows the fairness in the sense that the weighted share is nearly identical for each traffic flow with few variations. All the above indicates that the design goals, achieving weighted fairness and performance maximization can be actually obtained with our method with the P-IEEE protocol.

Figure 5 shows the delay results, with the same trend as that in Fig. 4. However, when comparing the two figures, we have more observations. First, the minimum frame delay is achieved when the maximum aggregation throughput is obtained, with the same $P_{i,j}^{t^*}$s either exactly or approximately[1]. This complies with our theoretical analysis and confirms the same trend in the single class case. Second, the fact that both optimal methods and approximations can produce similar results despite the scenarios implies that our algorithms can tolerate the variance of τ_1^* to some content. This robustness is particularly useful because having this property, our algorithms could be expected to produce stable results even if the value of τ_1^* may vary due to uncertain channel condition in WLANs.

[1] Note that, however, the $P_{i,j}^{t^*}$s involved are not shown in the figures.

6 Conclusion

In this paper, we use a p-persistent transmission control protocol to enhance the legacy IEEE 802.11 DCF with the capability of achieving weighted fairness among data traffic in different priority classes in WLANs. Experiment results indicate that with our method, the p-persistent enhanced MAC can actually achieve the weighted share, and also maximize the aggregate throughput and minimize the frame delay, which is a hard task for IEEE 802.11 WLANs even not impossible.

When compared with other complex or incompatible modifications to the IEEE 802.11 MAC, the proposed method has the characteristics of simplicity and complete distribution, and requires no extra messages to be shared among cooperating neighbor nodes. As a result, this method is considered as a more convenient alternate that can properly provide differential services in the WLANs, with the p-persistent transmission probability, P_T, as a parameter that can comply with the legacy DCF.

References

1. IEEE 802.11 II: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Standard, IEEE (August 1999)
2. Shih, C.-Y., Cheng, R.-G., Chang, C.-J.: Achieving Weighted Fairness for Wireless Multimedia Services. In: Proc. of IEEE VTC 2003, vol. 4, pp. 2381–2385 (October 2003)
3. Luo, H., Lu, S., Bharghavan, V.: A New Model for Packet Scheduling in Multihop Wireless Networks. In: Proc. of ACM MobiCom 2000, pp. 76–86 (2000)
4. Qiao, D., Shin, K.G.: Achieving Efficient Channel Utilization and Weighted Fairness for Data Communications in IEEE 802.11 WLAN under the DCF. In: Proc. of IWQoS'2002, pp. 227–236 (2002)
5. Ge, Y., Hou, J.: An Analytical Model for Service Differentiation in IEEE 802.11. In: Proc. of IEEE ICC 2003, vol. 2, pp. 1157–1162 (May 2003)
6. Cali', F., Conti, M., Gregori, E.: IEEE 802.11 Wireless LAN: Capacity Analysis and Protocol Enhancement. In: Proceeding of INFOCOM 1998, pp. 142–149 (March 29 - April 2, 1998)
7. Liu, J.S.: Design and Performance Evaluation of a Distributed Transmission Control Protocol for Wireless Local Area Network. IEICE Transactions on Communications E89-B(6) (June 2006)
8. Bianchi, G.: Performance Analysis of the IEEE 802.11 Distributed Coordination Function. IEEE Journal on Selected Areas in Communications 18(3), 535–547 (2003)
9. Vassis, D.E.: http://www.icsd.aegean.gr/telecom/Pythagor/

Dynamic Hierarchical Location Management Scheme for Host Identity Protocol*

Shuigen Yang, Yajuan Qin, and Dong Yang

Beijing Jiaotong University, 100044 Beijing, China
ipv6ysg@163.com, yjqin@bjtu.edu.cn, youngmanyd@sohu.com

Abstract. In this paper, a dynamic hierarchical location management scheme for Host Identity Protocol called DH-HIP is proposed to support micro-mobility. DH-HIP has a three-layer architecture which is managed by Rendezvous Server (RVS), Gate RVS and Local RVS (LRVS) respectively. The host selects its LRVS and computes the optimal size of administrative domain according to its current mobility and packet arrival rate. Furthermore, an analytical model to study the performance of DH-HIP and HIP is presented. Analytical results show that the signaling cost is significantly reduced through DH-HIP compared with the IETF HIP scheme under various conditions.

1 Introduction

The current fast increasing demand for wireless access to Internet applications is fueled by the remarkable access of wireless communication networks and the explosive growth of the Internet. Since the existing Internet was originally designed for communications between fixed nodes, Internet mobility support is a very complicated topic, and there are a lot of issues to be resolved. Since last decade, studies that address these issues have arisen, coming up with a number of protocol proposals, such as Mobile IP (MIP) [1] and Stream Control Transmission Protocol (SCTP) [2]. However, these solutions have some drawbacks, such as triangular routing and lack of location management and complexity [3].

In the traditional TCP/IP protocol stacks, the IP address suffers from semantic overloading by representing both the topological location and the identifier for a network interface [4]. This dual nature makes it difficult to enable mobility. In the traditional TCP/IP protocol stacks, a change of the IP address makes it impossible for other devices to contact the device using a constant IP address. In addition, even if the roaming device is able to obtain a new IP address dynamically, the transport connections established in the previous network will disconnect because the service is still bound to the old IP address, and then it has to re-establishes the connections.

Host Identity Protocol (HIP) [5] is designed to establish secure communication and to provide continuity of communication. It separates the identifier

* This research is supported by the National Science Foundation of China (No. 60573001, 60674008) and the China Next Generation Internet Project ([2006]2075).

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 185–196, 2007.

and locator roles of the IP address by introducing a new namespace, the Host Identity namespace, and a new layer between network layer and transport layer. The Host Identity namespace consists of Host Identity Tags (HIT) which are the public keys of public-private key pairs. The transport layer connections are no longer bound to the IP addresses but to HITs, and the IP address becomes pure routing message. Therefore, the changes of IP addresses are transparent to transport layer connections, and they do not have to be broken.

HIP can support macro-mobility, but it shows unnecessary signaling overhead and handoff latency when used in micro-mobility environment. In this paper, we develop a dynamic hierarchical location management scheme for HIP to make it capable to support micro-mobility situations.

The rest of the paper is organized as follows. Section 2 gives a short overview of HIP and some related woks. Section 3 describes the dynamic hierarchical location management scheme for HIP in detail. Section 4 develops an analytical model of the signaling costs of the proposed scheme and HIP. Section 5 shows the performance results. Finally, Section 6 concludes this paper.

2 Background

HIP defines a four-way handshake mechanism called HIP base exchange to establish a HIP layer end-to-end connection [6]. During the base exchange, a Diffie-Hellman procedure is used to create a session key and to establish a pair of IP Security (IPsec) Encapsulated Security Payload (ESP) Security Association (SA) between the two endpoints [7]. The message sequence of HIP base exchange is shown in Figure 1.

The Initiator begins the negotiation by sending an I1 packet which contains the HITs of the nodes participating in the negotiation. Then the Responder sends back a R1 packet that contains a puzzle to be solved by the Initiator. The R1 also initiates the Diffie-Hellman procedure, containing the public key of the Responder together with the Diffie-Hellman parameters. Once received the R1 packet, the Initiator solves the puzzle and sends theresponder cookie in an I2 packet together with an IPsec Security Parameters Index (SPI) value and its

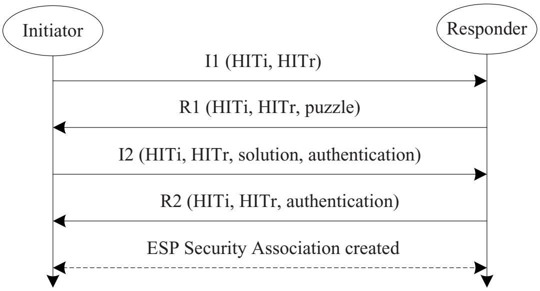

Initiator
Responder
I1 (HITi, HITr)
R1 (HITi, HITr, puzzle)
I2 (HITi, HITr, solution, authentication)
R2 (HITi, HITr, authentication)
ESP Security Association created

Fig. 1. HIP base exchange

encrypted public key to the Responder. The Responder verifies the solution and creates the IPsec ESP SAs. The final R2 message contains the SPI value of the Responder.

HIP introduces Rendezvous Server (RVS) to support mobility [8]. The Mobile Node (MN) which enters the network registers its IP address at RVS and reports the IP address of RVS at DNS. If the Correspondent Node (CN) wants to connect the MN, it performs a lookup at DNS. The DNS answers with the IP address of MN's RVS. The CN now initiates the connection by sending I1 packet to RVS with the HIT of MN. RVS adds a FROM parameter to I1 representing the IP address of CN, and forwards it to MN. When MN receives I1, it sends R1 directly to CN. The MN also adds a VIA_RVS parameter to R1, which contains the IP address of RVS. Finally, the MN and CN finish the HIP base exchange in the regular way. If the MN changes its IP address, it updates the relationship between its IP address and HIT at RVS and CN. This mechanism supports the simultaneous mobility of both nodes. However, it may create unnecessary signaling and control messages in micro-mobility environments.

Xie et al. [9] proposed a distributed dynamic regional location management for MIP. It can be considered as the extension of the IETF regional registration scheme to make it more flexible and adaptive. In this scheme, the system has a two-layer architecture which is managed by the Home Agent and Gateway FA respectively. However, Misra et al. [10] indicated that three-layer network architecture was the optimal hierarchical mobility management network architecture. Some schemes based on "pointer forwarding" technique were proposed in [11] and [12]. In these schemes, pointers are setup when MN moves to a new subnet. However, the authors assume there is correlation between the communication and the geographic distances and the MN's coordinate need to be known in the mobility management.

3 Dynamic Hierarchical Location Management

In this chapter, we introduce the proposed dynamic hierarchical location management scheme for HIP. We call this scheme DH-HIP. It is based on two new network entities called Gateway Rendezvous Server (GRVS) and Local Rendezvous Server (LRVS).

3.1 Architecture

Figure 2 shows the DH-HIP architecture. The network is divided into some autonomous domains, and each autonomous domain is divided into some administrative domains. For each administrative domain, there is a LRVS responsible for managing the MN's HIT and IP address. For each autonomous domain, there is a GRVS responsible for managing the relationship between the LRVS and the MN. The GRVS and LRVS extend the functionalities of the RVS. The LRVS knows the HIT and IP address of the GRVS which it attaches to.

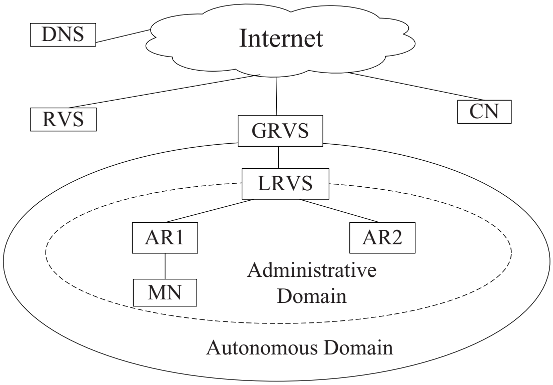


Fig. 2. DH-HIP architecture

Under DH-HIP, each Access Router (AR) can function either as an AR or a LRVS depending on the user mobility. If the AR acts as a LRVS, it maintains a visitor list and keeps entries in the list update according to the registration request sent from MNs. LRVS also relays all the autonomous registration requests to GRVS and home registration requests to RVS.

When MN enters the network, it registers its HIT and IP address at LRVS, GRVS and RVS respectively. Firstly, the MN registers its HIT and IP address at LRVS, and LRVS records the HIT and IP address of the MN. Secondly, the MN registers its HIT and IP address at GRVS. During this procedure, the MN sends the registration packets to GRVS. When LRVS intercepts these packets, it changes the IP address of the MN in the header to the IP address of the LRVS, and forwards them to GRVS. Therefore, the GRVS records the HIT of MN and the IP address of the LRVS. Thirdly, the MN registers its HIT and IP address at RVS. During this procedure, the MN sends registration packets to RVS. When GRVS intercepts these packets, it changes the IP address of the MN in the header to the IP address of the GRVS, and forwards them to RVS. After this registration, RVS records the HIT of the MN and the IP address of the GRVS.

If CN wants to communicate with MN, it queries at DNS. The DNS answers with the IP address of the RVS. Then CN sends I1 of the HIP base exchange to RVS. When RVS receives I1, it adds a FROM_CN parameter to it, and forwards it to GRVS within the given autonomous domain. The GRVS adds a FROM_RVS parameter to I1 and forwards it to LRVS within the given administrative domain. And then LRVS adds a FROM_GRVS parameter to I1 and forwards it to MN. When MN receives I1, it adds VIA_RVS, VIA_GRVS and VIA_LRVS parameters to R1, and then sends it to LRVS. LRVS forwards it to CN. The remained HIP base exchange (I2, R2) continues in the regular way, without including the RVS and GRVS, but with the forwarding of LRVS.

Figure 3 depicts the message flows for the connection setup.

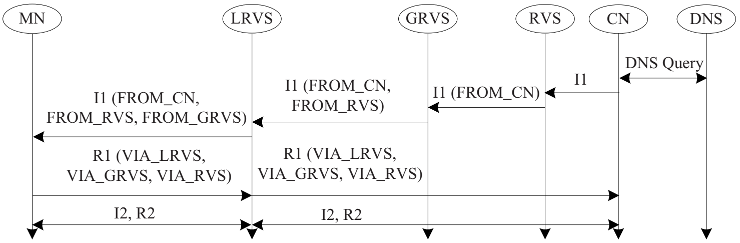

MN
LRVS
GRVS
RVS
CN
DNS
DNS Query
I1
I1 (FROM_CN)
I1 (FROM_CN, FROM_RVS)
I1 (FROM_CN, FROM_RVS, FROM_GRVS)
R1 (VIA_LRVS, VIA_GRVS, VIA_RVS)
R1 (VIA_LRVS, VIA_GRVS, VIA_RVS)
I2, R2
I2, R2

Fig. 3. Message flows for the registration and connection setup

3.2 Dynamic Location Management

Under DH-HIP, the number of ARs under a LRVS is not fixed but optimized for each MN to minimize the total signaling cost. The optimal number is obtained based on the packet arrival rate and mobility rate of MN. Since the packet arrival and mobility rate of each MN are different and they may not be constant from time to time, the optimal number of ARs is different for each MN and it is adjustable from time to time.

Each MN keeps a buffer for storing HITs and IP addresses of the ARs and GRVS. If the MN enters the network, it connects to one of the ARs directly. The AR sends advertisement messages which contain its HIT and IP address. They also contain the HIT and IP address of the GRVS which is responsible for the given autonomous domain. These advertisement messages are modified Internet Control Message Protocol version 6 (ICMPv6) Router Advertisement messages. The first AR the MN visits will function as the LRVS of the administrative domain. The MN registers its HIT and IP address at LRVS, GRVS and RVS respectively. Then it computes the optimal size of this administrative domain based on its packet arrival rate and mobility rate. This optimal value k_{opt} is set for the buffer length threshold of the MN.

When MN detects that it enters a new subnet, it compares the GRVS IP address in the new subnet which obtained from the advertisement messages with the IP addresses recorded in its buffer. If the current GRVS IP address has not been recorded or it is not the same as its records in the buffer which means the MN has moved to a new autonomous domain, the MN deletes all the HITs and IP addresses in its buffer and records the HITs and IP addresses of the new AR and GRVS. The MN selects the new AR as LRVS of the new administrative domain. Then the MN registers its HIT and IP address at the new LRVS and GRVS, updates its registration at RVS and CN. The MN also computes k_{opt} for the new administrative domain.

If the current GRVS IP address is the same as its records in the buffer which means the MN moves within the same autonomous domain, the MN compares the current AR's IP address with the IP addresses recorded in its buffer. If the current AR's IP address has not been recorded, the MN records it. Otherwise, ignores it and updates its registration at LRVS and CN. If the total number of IP addresses in the buffer as well as the current AR's IP address exceeds k_{opt}, which means the MN is in a new registered domain, the MN deletes all the HITs and IP addresses of ARs and LRVS in the buffer, saves the new one, registers its HIT and IP address at the current AR, and updates its registration at GRVS and CN. If the total number of IP addresses in the buffer as well as the current AR IP address does not exceed k_{opt}, it means the MN is in the same administrative domain. Then the MN updates its registration at LRVS and CN.

4 Signaling Cost Function

In this section, we derive the cost function of location update and packet delivery. The total signaling cost in location update and packet delivery is considered as the performance metric. The model parameters are defined as follows.

c_{ma}	The transmission cost between MN and AR.
c_{al}	The transmission cost between AR and LRVS.
c_{gl}	The transmission cost between GRVS and LRVS.
c_{gr}	The transmission cost between GRVS and RVS.
c_{ar}	The transmission cost between AR and RVS.

4.1 Location Update Cost

The location update cost at CN is just the same under HIP and DH-HIP. Therefore, we do not consider the location update cost at CN.

Let a_m and a_a denote the registration processing cost at MN and AR respectively. Let a_l, a_g , and a_r denote the location update and registration processing cost at LRVS, GRVS and RVS respectively.

Assume the packet arrivals to an MN form a Poisson process with arrival rate λ, and the time MN resides in a subnet has a mean $\frac{1}{\mu}$. Define the Call-to-Mobility Ratio (CMR) as follows.

$$\rho = \frac{\lambda}{\mu} \tag{1}$$

where the ρ is the CMR. According to the procedures of location update and registration in Section 3, the location update cost under HIP (U_{HIP}) is

$$U_{HIP} = \frac{a_r + 2a_a + 2c_{ma} + 2c_{ar}}{\rho} \tag{2}$$

When MN moves within the same administrative domain, it updates its IP address at LRVS under DH-HIP. The location update cost at LRVS (U_l) is

$$U_l = a_l + 2a_a + 2c_{ma} + 2c_{al} \tag{3}$$

When MN moves between different administrative domains which belong to the same autonomous domain, it selects the first AR as its new LRVS, registers its HIT and IP address at the new LRVS and updates its IP address at GRVS.

The registration cost at the new LRVS (U_{g1}) and the location update cost at GRVS (U_{g2}) are

$$U_{g1} = a_m + 2a_l + 4c_{ma} \tag{4}$$

$$U_{g2} = a_g + 2a_l + 2c_{ma} + 2c_{gl} \tag{5}$$

Therefore, the total location update cost under DH-HIP when MN moves between different administrative domains (U_g) is

$$U_g = U_{g1} + U_{g2} = a_m + a_g + 4a_l + 6c_{ma} + 2c_{gl} \tag{6}$$

When MN moves between different autonomous domains, it registers its HIT and IP address at the new LRVS and GRVS, and updates its IP address at RVS. The registration cost at the new LRVS (U_{r1}) and at the new GRVS (U_{r2}), and the location update cost at RVS (U_{r3}) are

$$U_{r1} = a_m + 2a_l + 4c_{ma} \tag{7}$$

$$U_{r2} = a_m + 2a_g + 4a_l + 4c_{ma} + 4c_{gl} \tag{8}$$

$$U_{r3} = a_r + 2a_g + 2a_l + 2c_{ma} + 2c_{gl} + 2c_{gr} \tag{9}$$

Therefore, the total location update cost under DH-HIP when it moves between different autonomous domains (U_r) is

$$U_r = a_r + 2a_m + 4a_g + 8a_l + 10c_{ma} + 6c_{gl} + 2c_{gr} \tag{10}$$

Let a denote the number of subnets crossed between the last packet arrival and the very last location update just before the last packet arrival. Its probability distribution is $\beta(a)$. Let i denote the number of subnets crossed between two consecutive packet arrivals. Its probability distribution is $\alpha(i)$. Let m and k denote the number of subnets within an autonomous domain and an administrative domain respectively.

Under DH-HIP, the MN will update to GRVSLRVS and RVS $\lfloor \frac{i+a}{k} \rfloor$, $\lfloor \frac{i+a}{m} \rfloor$ and $i - \lfloor \frac{i+a}{k} \rfloor - \lfloor \frac{i+a}{m} \rfloor$ times respectively. The total location update cost under DH-HIP (U_{DH-HIP}) is

$$U_{DH-HIP} = \sum_{i=0}^{\infty} \{ \lfloor \frac{i+a}{k} \rfloor U_g + \lfloor \frac{i+a}{m} \rfloor U_r + (i - \lfloor \frac{i+a}{k} \rfloor - \lfloor \frac{i+a}{m} \rfloor) U_l \} \alpha(i) \tag{11}$$

Assume the residence time of an MN in a subnet is a random variable with a general density function $f(t)$ and the Laplace transform is $f^*(s) = \int_{t=0}^{\infty} f(t) e^{-st} dt$. Denote $g = f^*(\lambda)$, then $\alpha(i)$ is given by [13]

$$\alpha(i) = \begin{cases} 1 - \frac{1-g}{\rho}, & \text{if } \ i = 0 \\ \frac{(1-g)^2 g^{i-1}}{\rho}, & \text{if } \ i > 0 \end{cases} \tag{12}$$

Substitute $i = jmk + q$. Notice that both $0 \leq q < k$ and $0 \leq a < k$, then

$$\begin{aligned}
U_{DH-HIP} &= \frac{U_l}{\rho} + \sum_{j=0}^{\infty}\sum_{q=0}^{k-1}\{\lfloor\frac{jmk+q+a}{k}\rfloor(U_g - U_l)\}\frac{(1-g)^2}{\rho g}g^{jmk+q} \\
&\quad + \sum_{j=0}^{\infty}\sum_{q=0}^{k-1}\{\lfloor\frac{jmk+q+a}{m}\rfloor(U_r - U_l)\}\frac{(1-g)^2}{\rho g}g^{jmk+q} \\
&= \frac{U_l}{\rho} + \sum_{j=0}^{\infty}\sum_{q=0}^{k-a-1}[jm(U_g - U_l) + jk(U_r - U_l)]\frac{(1-g)^2}{\rho g}g^{jmk+q} \\
&\quad + \sum_{j=0}^{\infty}\sum_{q=k-a}^{k-1}[(jm+1)(U_g - U_l) + (jk+1)(U_r - U_l)]\frac{(1-g)^2}{\rho g}g^{jmk+q} \\
&= \frac{U_l}{\rho} + \frac{(1-g)(g^{k-a} - g^k)}{\rho g(1-g^{mk})}(U_g + U_r - 2U_l) \\
&\quad + \frac{g^{mk}(1-g)(1-g^k)}{\rho g(1-g^{mk})^2}[m(U_g - U_l) + k(U_r - U_l)] \qquad (13)
\end{aligned}$$

4.2 Packet Delivery Cost

Similar to the analysis of packet delivery cost in [9], we consider the packet delivery cost for HIP and DH-HIP. Let ν_l, ν_g and ν_r denote the packet processing cost at LRVS, GRVS and RVS respectively.

Under HIP, the packet delivery cost (P_{HIP}) includes the packet delivery processing cost at RVS and the transmission cost between AR and RVS.

$$P_{HIP} = \nu_r + c_{ar} \qquad (14)$$

Assume there are ω MNs in a subnet. Under DH-HIP, the processing cost at LRVS includes checking its visitor list to see whether it has an entry for the destination MN and management of routing packets to AR. Since location database lookup is based on the traditional *Patricia trie*, the complexity of location database lookup is analogous to its length k. Therefore, the packet processing cost at GRVS for DH-HIP is

$$\nu_l = \xi k[\alpha\omega + \beta \log(k)] \qquad (15)$$

where ξ is a constant which captures the bandwidth allocation cost at LRVS, α and β are weighting factors of visitor list and location database lookups.

Under DH-HIP, the packet delivery cost (P_{DH-HIP}) includes the packet delivery processing cost at RVSGRVS and LRVS, the packet transmission cost between AR and LRVS, the packet transmission cost between LRVS and GRVS, and the packet transmission cost between GRVS and RVS.

$$P_{DH-HIP} = \nu_r + \nu_g + c_{gr} + c_{gl} + c_{al} + \xi k[\alpha\omega + \beta \log(k)] \qquad (16)$$

4.3 Optimal Threshold

The total signaling cost for DH-HIP (S_{DH-HIP}) and HIP (S_{HIP}) are

$$S_{DH-HIP} = U_{DH-HIP} + P_{DH-HIP} \tag{17}$$

$$S_{HIP} = U_{HIP} + P_{HIP} \tag{18}$$

The optimal threshold k_{opt} for an MN is defined as the value of k that minimizes the cost function derived above. Since the k_{opt} must be an integer, we use a similar method proposed in [9] to obtain the optimal value. Define the cost difference function as

$$\varDelta S(k, \rho) = S_{DH-HIP}(k, \rho) - S_{DH-HIP}(k-1, \rho) \tag{19}$$

where $k \geq 2$. Given the CMR, the algorithm to find the optimal threshold is defined as

$$k_{opt} = \begin{cases} 1, & \text{if } \varDelta S(2, \rho) > 0 \\ \max\{k : \varDelta S(k, \rho) \leq 0\}, & \text{otherwise} \end{cases} \tag{20}$$

5 Analytical Results

In this section, we present analytical results results showing the effect of various input parameters on total signaling cost. For numerical calculations, we use the following parameter values: $c_{ma} = 1$, $c_{al} = 2$, $c_{gl} = 5$, $c_{gr} = 10$, $c_{ar} = 15$, $a_m = 5$, $a_a = 10$, $a_l = 15$, $a_g = 25$, $a_r = 50$, $\nu_g = 12.5$, $\nu_r = 25$, $m = 30$, $\xi = 0.01$, $\alpha = 0.3$, $\beta = 0.7$, $\omega = 15$.

Assume that the subnet residence time is exponentially distributed, then

$$g = \frac{1}{1+\rho} \tag{21}$$

If a is uniformly distributed with $\beta(a) = \frac{1}{k}$, then

$$\begin{aligned} S_{DH-HIP} = P_{DH-HIP} + \frac{U_l}{\rho} + \frac{1-g^k}{\rho g k(1-g^{mk})}(U_g + U_r - 2U_l) \\ + \frac{g^{mk}(1-g)(1-g^k)}{\rho g(1-g^{mk})^2}[m(U_g - U_l) + k(U_r - U_l)] \end{aligned} \tag{22}$$

If a is linearly distributed with $\beta(a) = \frac{2(k-a)}{k(k+1)}$, then

$$\begin{aligned} S_{DH-HIP} = P_{DH-HIP} + \frac{U_l}{\rho} + \frac{g^{mk}(1-g)(1-g^k)}{\rho g(1-g^{mk})^2}[m(U_g - U_l) + k(U_r - U_l)] \\ + \frac{U_g + U_r - 2U_l}{\rho(k+1)(1-g^{mk})}[(1-k)(1-g)g^{k-1} + \frac{2g^{k-1}}{k}] \\ + \frac{U_g + U_r - 2U_l}{\rho(k+1)(1-g^{mk})} \cdot \frac{2(1-g^{k-1})}{k(1-g)} \end{aligned} \tag{23}$$

If a is exponentially distributed with $\beta(a) = \frac{(1-e^{-1})e^{-a}}{1-e^{-k}}$, then

$$S_{DH-HIP} = P_{DH-HIP} + \frac{U_l}{\rho} + \frac{g^{mk}(1-g)(1-g^k)}{\rho g(1-g^{mk})^2}[m(U_g - U_l) + k(U_r - U_l)]$$
$$+\frac{(1-g)(1-e^{-1})g^k}{\rho g(1-e^{-k})(1-g^{mk})} \cdot \frac{1-ge^{1-k}}{ge-1}(U_g + U_r - 2U_l)$$
$$+\frac{(1-g)(1-e^{-1})g^k}{\rho g(1-e^{-k}(1-g^{mk})} \cdot \frac{1-e^{1-k}}{1-e}(U_g + U_r - 2U_l) \qquad (24)$$

5.1 Impact of CMR

Figure 4, Figure 5 and Figure 6 shows the total signaling cost as a function of CMR for HIP and DH-HIP under uniform distribution, linear distribution and exponential distribution respectively.

As shown in Figure 4, Figure 5 and Figure 6, the total signaling cost decreases as CMR increases. When CMR is low, the mobility rate is high compared to the packet arrival rate and the cost for location update dominates. Systems with larger administrative domain may reduce the number of registrations at GRVS and RVS. Therefore, the DH-HIP scheme generates less signaling cost than the HIP scheme. When CMR is high, the packet delivery cost becomes significant. The saving can be attributed to the smaller administrative domain.

When CMR=0.1, the mobility rate is high compared to the packet arrival rate, and the MN will choose large threshold k_{opt} to reduce the total signaling cost. Under uniform distribution, DH-HIP can reduce 44.09% signaling cost. Under linear distribution, DH-HIP can reduce 50.14% signaling cost. Under exponential distribution, DH-HIP can reduce 55.83% signaling cost.

When CMR=1, the mobility rate is the same as the packet arrival rate. Under uniform distribution, DH-HIP can reduce 13.98% signaling cost. Under linear distribution, DH-HIP can reduce 29.97% signaling cost. Under exponential distribution, DH-HIP can reduce 32.72% signaling cost.

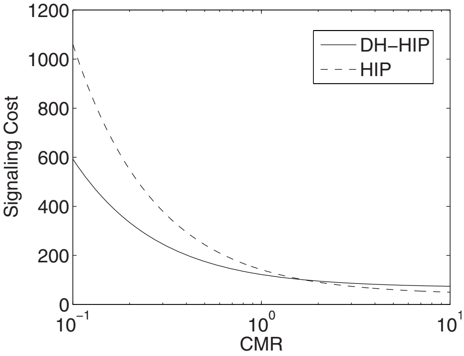


Fig. 4. Comparison of the total signaling cost under uniform distribution

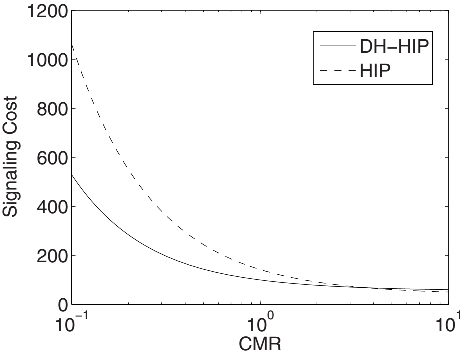


Fig. 5. Comparison of the total signaling cost under linear distribution

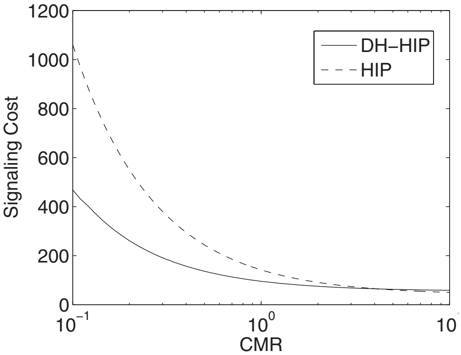


Fig. 6. Comparison of the total signaling cost under exponential distribution

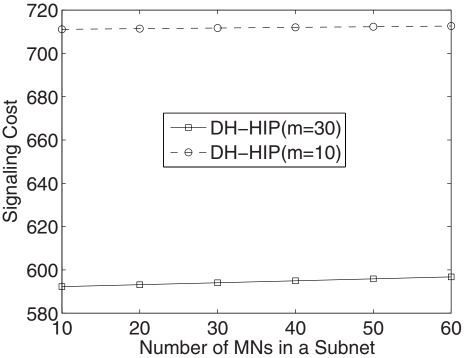


Fig. 7. Total signaling cost as a function of ω for DH-HIP under uniform distribution

However, when CMR is high, DH-HIP will produce more signaling cost than HIP. This is due to the large packet delivery cost at LRVS under DH-HIP. Consider the signaling cost for DH-HIP and HIP under CMR=10. Under uniform distribution, DH-HIP will increase 48.06% signaling cost. Under linear distribution, DH-HIP will increase 19.43% signaling cost. Under exponential distribution, DH-HIP will increase 16.82% signaling cost.

5.2 Impact of the Number of MNs

Another important measurement of performance is the number of MNs in a subnet. We fixed CMR=0.1, and ω varies from 10 to 60. Figure 7, Figure 8 and Figure 9 show the total signaling cost as a function of ω for DH-HIP when $m = 30$ and $m = 10$ under uniform distribution, linear distribution and exponential distribution respectively.

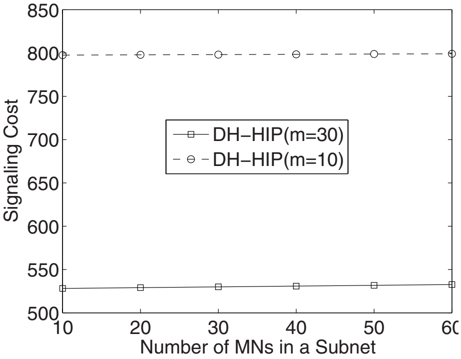


Fig. 8. Total signaling cost as a function of ω for DH-HIP under linear distribution

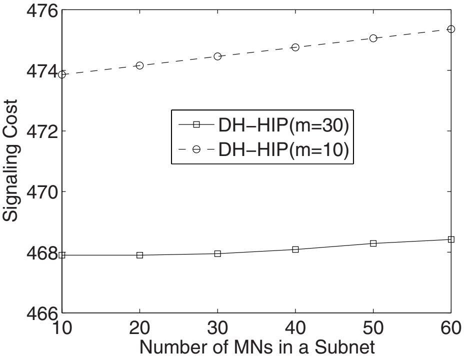


Fig. 9. Total signaling cost as a function of ω for DH-HIP under exponential distribution

As shown in Figure 7, Figure 8 and Figure 9, the signaling cost for DH-HIP increases slowly as the number of MNs increases. Consider the signaling cost for DH-HIP and HIP when $\omega = 10$ and $\omega = 60$. Under uniform distribution, DH-HIP only increases 0.76% and 0.21% signaling cost when $m = 30$ and $m = 10$ respectively. Under linear distribution, DH-HIP only increase 0.85% and 0.19% signaling cost when $m = 30$ and $m = 10$ respectively. Under exponential distribution, DH-HIP only increase 0.11% and 0.32% signaling cost when $m = 30$ and $m = 10$ respectively.

6 Conclusion

In this paper, we introduced a dynamic hierarchical location management scheme for HIP. Under this scheme, the system has a three-layer architecture, and the MN dynamically optimizes the administrative domain size according to its current mobility rate and packet arrival rate. Analytical results show that the signaling cost is significantly reduced through our proposed dynamic hierarchical system architecture compared with the IETF HIP scheme under various conditions.

References

1. Perkins, C.: IP Mobility Support for IPv4. IETF RFC 3220 (2002)
2. Stewart, R., Xie, Q., Morneault, K.: Stream Control Transmission Protocol. IETF RFC 2960 (2000)
3. Henderson, T.R.: Host Mobility for IP Networks: A Comparison. IEEE Network 17(6), 18–26 (2003)
4. Balakrishnan, H., Lakshminarayanan, K., Ratnasamy, S.: A Layered Naming Architecture for the Internet. ACM SIGCOMM Computer Communication Review 34(4), 343–352 (2004)
5. Moskowitz, R., Nikander, P.: Host Identity Protocol (HIP) Architecture. IETF RFC 4423 (2006)
6. Moskowitz, R., Nikander, P., Jokela, P.: Host Identity Protocol. Internet draft-ietf-hip-base-07 (2007)
7. Jokela, P., Moskowitz, R., Nikander, P.: Using ESP Transport Format with HIP. Internet draft-ietf-hip-esp-05 (2007)
8. Laganier, J., Eggert, L.: Host Identity Protocol (HIP) Rendezvous Extension. Internet draft-ietf-hip-rvs-05 (2006)
9. Xie, J., Akyildiz, I.F.: A Distributed Dynamic Regional Location Management Scheme for Mobile IP. In: INFOCOM 2002, pp. 1069–1078. IEEE Press, New York, USA (2002)
10. Misra, I.S., Chakraborty, M.: An Approach for Optimal Hierarchical Mobility Management Network Architecture. In: VTC 2006-Spring, pp. 481–485. IEEE Press, Melbourne, Australia (2006)
11. Bejerano, Y., Cidon, I.: An Anchor Chain Scheme for IP Mobility Management. Wireless Network 9(5), 409–420 (2003)
12. Chu, C.H., Weng, C.M.: Pointer Forwarding MIPv6 Mobility Management. In: GLOBECOM 2002, pp. 2133–2137. IEEE Press, Taipei, Taiwan (2002)
13. Fang, Y., Chlamtac, I., Lin, Y.B.: Portable Movement Modeling for PCS Networks. IEEE Transactions on Vehicular Technology 49(4), 1356–1363 (2000)

Migration Policies for Location-Centric Data Storage in Mobile Ad-Hoc Networks

Dominique Dudkowski[1], Pedro José Marrón[2], and Kurt Rothermel[1]

[1] University of Stuttgart, 70569 Stuttgart, Germany,
{dudkowski|rothermel}@ipvs.uni-stuttgart.de
[2] University of Bonn, 53117 Bonn, Germany,
pjmarron@cs.uni-bonn.de

Abstract. *Location-centric* data storage is a fundamental paradigm for data management in wireless ad-hoc networks. It guarantees that data is stored at network nodes near specific geometric reference locations in the region where the network is deployed. In mobile ad-hoc networks, maintaining spatial proximity between data and its associated location requires explicit migration mechanisms in order to "keep the data in place". In this paper we propose comprehensive policies for data migration that effectively maintain the spatial coherence of data given the particular characteristics of mobile ad-hoc networks. Using extensive simulations we show how the proposed policies outperform related migration approaches over a wide range of system parameter settings, in particular, node density, network dynamics, and migratable data size.

1 Introduction

Data management in wireless ad-hoc networks has become a prolific field of research and significant progress has been achieved in storing and processing data in these networks. A considerable number of algorithms, most prominently location and spatial query services [8], [14], [3] as well as distributed and geographic hash tables [11], [6] [12], [9], share a common feature: they frequently rely on data being stored near certain geometric locations (points or regions) in the area where the network is deployed. We will refer to this kind of storage paradigm throughout this paper as *location-centric* data storage (LCDS).

Data stored according to the LCDS approach can be efficiently retrieved by making use of suitable geometric routing protocols. For example, in the geographic hash tables implementation for wireless sensor networks [11], individual data items (events) are first hashed to geometric coordinates. Then, LCDS works by using customized greedy perimeter stateless routing [7], which deterministically routes read and write requests to the nodes storing the relevant data.

In this paper, we focus on mobile ad-hoc networks (MANETs). Data stored on single network nodes becomes itself mobile when a storage node is moving and, thus, carries the data along with it. To retain spatial coherence, that is, the proximity of data to its associated location, it is necessary to migrate data from its current storage node to another more suitable one.

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 197–208, 2007.

Due to their individual storage requirements, the aforementioned algorithms implement specific solutions to maintain spatial coherence, which makes them inappropriate for being used in a more flexible LCDS system. Two of these issues concern the migratable data size and the handling of significant node mobility, and we will discuss related requirements in more detail in Sec. 2.

The contribution of this paper is, therefore, the presentation of comprehensive strategies for data migration under the specific characteristics of MANETs, which may exhibit poor topological connectivity for low node densities and tough mobility patters. More specifically, we propose a comprehensive migration policy framework that determines the necessity of migrations and their potential benefit in terms of achieving spatial coherence. Our policies provide the necessary means to choose suitable migration target nodes while at the same time taking into account the stability of the network topology utilized during migration.

The remainder of the paper is structured as follows. In the next section, we discuss related work. The system model is given in Sec. 3, followed by a description of the proposed migration policies in Sec. 4. Integrated with the storage approach taken from [4], we will validate policy performance in Sec. 5 before we conclude the paper with a summary and further issues in Sec. 6.

2 Related Work

In the context of our work, we focus on contributions relevant to the specific field of LCDS in MANETs. We have identified the following primary requirements that effective migration policies must take into account:

1. Resilience to significant node mobility (e.g. vehicular ad-hoc networks)
2. Independence from assumptions that are based on node density
3. Resilience to weak connectivity of the network topology
4. Tolerance to occasional node and frequent communication failures
5. Support for large quantities of (dynamic) migratable data
6. Genericity for seamless integration with existing storage protocols
7. Possibility to (dynamically) control the level of spatial coherence

Migration functionality is adopted by a number of distributed hash tables (DHTs) for MANETs (those for stationary networks are omitted). Rendezvous Regions [12] and Cell Hash Routing [1] implement region-based LCDS. When servers move out of a region, data handovers attempt to retain the data on other servers in the respective region. Because these approaches rely on fixed-sized geometric areas, they depend on sufficient local node population for proper operation of handovers. Thus, the approaches are in violation with our requirements 2 and 3. In addition, data is dispersed over a potentially large number of servers inside of a region, which makes generic data management (e.g. replication) significantly more complex (requirement 6). In the DHT approach proposed in [9] a server migrates data to another node upon reaching a given threshold distance. To ensure that nodes exist near given reference coordinates the approach assumes sufficient node population, thus also violates requirement 2.

The authors of [13] provide an approach to data migration in the context of location-specific data retrieval (LCDS in our terms). However, the approach also heavily relies on geometric regions (target and expected zone) and thus, does not meet requirement 2. While the approach considers data handoff criteria that are local to a data server, target node selection is not considered. However, choosing a suitable target node is essential to satisfy requirements 1, 3, and 5.

Several location and query services also employ cell-based structures that eventually require the handover of position information. We identified [8] and [14] to be closest to our work. However, these approaches are vulnerable to the previously discussed requirements. In [4] the authors propose a novel LCDS approach based on the server abstraction and which does not rely on assumptions based on node density, thus meeting requirements 2, 3 and 6. The same authors propose algorithms for spatial queries in [3] that rely on servers located inside of geometric cells, making the approach vulnerable to requirement 2. Both approaches leave the design of comprehensive migration policies to future work.

Apart from the specific shortcomings of each discussed approach, requirements 1, 4, 5 and 7 are not addressed to a sufficient extent. To the best of our knowledge, comprehensive migration policies that enable efficient and resilient data migration for location-centric data storage in mobile ad-hoc networks that meet all of the aforementioned requirements is an open issue.

3 System Model

For the purposes of this paper, we consider a mobile ad-hoc network deployed inside of a Euclidean 2-space. Network nodes u_i communicate via wireless links, characterized by the nominal transmission range r_{tx}. We do not assume any particular propagation model, i.e. the effective transmission range of nodes may vary from r_{tx} with time. Let $\mathbf{r}_i(t)$ denote the trajectory of node u_i. We put no restrictions on how coordinates are obtained, both physical (e.g. WGS84 through GPS) or virtual coordinates (e.g. [10]) are supported by our model.

Let D denote a set of data objects $o_i \in D$ to be stored in the network. Let C denote a set of *reference coordinates* $\mathbf{c}_r \in C$. By $R \in D \times C$ we denote a (possibly dynamic) relation between data objects and reference coordinates, known to all nodes. Relation R induces *data subsets* $D(\mathbf{c}_r) \subseteq D$, containing data objects related to the same reference coordinate. A network node u_i is said to be the *data server* for data subset $D(\mathbf{c}_r)$ if it is responsible for storing all data objects of that subset. A data server may manage several data subsets at the same time. By definition and w.l.g., a data subset denotes the smallest migratable unit.

4 Data Migration Policies

The task of the migration policies is to enforce spatial coherence between a data subset and the reference coordinate the subset is associated with. Periodically, storage nodes are recruited that are located in close proximity to the relevant reference coordinate. This section details the migration policies that determine the necessity, benefit and success of a potential migration. We discuss aspects of integrating the policies with the LCDS approach in [4] in Sec. 5.

Our approach works by splitting the migration policy in two parts: the migration recommendation and the migration decision. The *migration recommendation policy* (MRP, Sec. 4.1) uses the **local state** of the server to determine whether migration is needed and beneficial, that is, leads to an increase in LCDS efficiency. The *migration decision policy* (MDP, Sec. 4.2) is invoked only if the MRP recommends that migration should be carried out. It examines the **remote state** of a set of possible candidate target nodes and attempts to find the most eligible one. The MDP also considers the network stability of the current topology in order to determine whether or not a sufficiently stable path exists to perform a successful migration. Finally, a migration is only performed if both the MRP and MDP agree that migration should take place.

4.1 Migration Recommendation Policy

Each data server monitors in regular time intervals if the MRP ought to be invoked for a particular data subset. After each period, the MRP examines the local state and evaluates the *migration recommendation predicate*, denoted P_{MRP}. If it is true, local state indicates that migration is necessary from the point of view of the current data server, and that it will most likely increase spatial coherence, and thus, LCDS efficiency. If $P_{\text{MRP}} = \text{false}$, the evaluation of the MRP is deferred to the next monitoring cycle. The length of the monitoring cycle is determined by the application and is a configurable parameter.

Besides this predicate, we introduce the notion of *migration priority* $\mathit{\Pi} \in (0, 1)$ that associates the MRP with the MDP. This priority value dictates the strictness by which the subsequent MDP must find a target server. If $\mathit{\Pi} = 0$ (minimum priority), the MDP has full freedom on choosing a candidate to receive the migrated data. If $\mathit{\Pi} = 1$ (maximum priority), the MDP is forced to find a target node, even in the case where no good candidate can be found.

Let us first consider predicate P_{MRP}, which ensures that a data server will migrate when it moves too far away from its associated reference coordinate. Let t_0 denote the current time, $\mathbf{r}_0(t_0)$ the position of the server at t_0, and $d_0 = d_0(t_0) = |\mathbf{r}_0(t_0) - \mathbf{c}_r|$.

Several distances are used to reason about how critical a migration is from the point of view of spatial considerations. By d_{opt}, we denote the optimal distance to the reference coordinate. Nodes located between d_{thresh} and d_{crit} are still suitable candidates. Beyond d_{crit}, nodes are considered no longer adequate to take the role of a data server. We choose $d_{\text{opt}}, d_{\text{thresh}}, d_{\text{crit}}$ to be multiples of r_{tx} but this is not strictly necessary. Using the threshold distance, the migration recommendation predicate is

$$P_{\text{MRP}} = \text{true} \Leftrightarrow d_0 > d_{\text{thresh}} \tag{1}$$

In other words, a server first considers a migration if it is farther away from the reference coordinate than d_{thresh}. This parameter is essential for the degree of spatial coherence that is desired in a particular scenario. That is, depending on the frequency of requests, this parameter can be used to fine-tune overall network performance. While a detailed analysis of this fact is beyond the scope of this paper, we show in Sec. 5 that spatial coherence is effectively maintained. If P_{MRP} is true, the migration decision policy will be triggered. Otherwise, migration is deferred to the next migration cycle.

When a server moves beyond d_{crit}, it will indicate that a migration *must* occur, since sufficient spatial coherence is no longer given. In that case, the MDP should not have any freedom to decide whether or not to postpone migration. Since the loss of spatial coherence is a gradual process, it is possible to design the following linear function for the migration priority Π:

$$\Pi = \begin{cases} 1 & \text{if} \quad d_0 > d_{\mathrm{crit}} \\ \frac{d_0 - d_{\mathrm{thresh}}}{d_{\mathrm{crit}} - d_{\mathrm{thresh}}} & \text{if} \quad d_{\mathrm{crit}} \geq d_0 \geq d_{\mathrm{thresh}} \\ 0 & \text{if} \quad d_0 < d_{\mathrm{thresh}} \end{cases} \tag{2}$$

With increasing Π, the MDP should decide on a migration more urgently. We will show in Sec. 4.2 how the priority influences the selection of possible target nodes consistent with the migration recommendation predicate in (1).

Apart from the spatial dimension, we have developed additional notions in the temporal domain to model load-balancing issues. These notions are particularly important when there are also stationary or slow-moving nodes in the network. In the following, we will assume that a migration will be initiated i) if a node wishes to log off the network explicitly (Sec. 3) or ii) after a finite time interval t_{thresh} if the migration predicate does not trigger.

4.2 Migration Decision Policy

The task of the migration decision policy (MDP) is to issue the final decision on whether to perform a migration once the MRP has recommended it, and to select a suitable target server u_T for the data subset to be migrated. In addition, the MDP determines an initial network path P_T for migration. In order to rate the eligibility of a possible candidate node u_i, we introduce the notion of *node eligibility* $\epsilon_i \in (0, 1)$. Complementary to the spatial considerations of the MRP, node eligibility is built on definitions that consistently integrate both policies. At the same time, we will incorporate migration priority Π to relax node selection if a sufficiently eligible node cannot be determined. Apart from node eligibility, the topological properties of the network must be taken into account to avoid starting a migration if the topology is not sufficiently stable. In this paper, we will focus on node eligibility and only briefly show how stability is considered using notions of the lifetime of network paths applied in previous work.

In order to obtain the necessary information prior to MDP evaluation, we execute a two-phase distributed collection algorithm. In the first phase, we flood request packets to a relevant subset of nodes and set up a reverse aggregation tree. Each node reached replies with information about its own local state relevant to the MDP via the previously built aggregation tree. The flooding region, denoted $\mathbf{A}_{\mathrm{MDP}}$, is defined in a way to reach those nodes that are most eligible in terms of spatial properties. Due to the lack of space, we omit the details of how we determine the shape of the region, as well as the precise collection algorithm. However, we note that no reliable communication is required, and information returned on a best-effort basis is fully sufficient for evaluation of the MDP.

In the following, we assume that an instance of state exploration was executed, and nodes u_i inside of $\mathbf{A}_{\mathrm{MDP}}$ have reported i) their geometric position $\mathbf{r}_i(t_i)$ at fixing time t_i, ii) their estimated speed $\bar{v}_i(t_i)$, and iii) the fixing time itself. By u_0, $\mathbf{r}_0(t_0)$, and $\bar{v}_0$ we denote the current server, its position at t_0, and its estimated speed, respectively.

We now deduce eligibility $\epsilon_i \in (0, 1)$ of each node u_i. Again, the eligibility of a node is generally larger when it is located closer to the reference coordinate, which is directly related to the MRP. An additional magnitude not present in the MRP is the fact that a node is more eligible when it is likely to move at low speed, therefore remaining in proximity to the reference coordinate for a longer period of time. Thus, we will characterize spatial eligibility on two dimensions, one being distance-based eligibility and the other sojourn-based eligibility.

Let us first characterize distance-based eligibility, and estimate the benefit that some node u_i relative to the current server u_0 will have in terms of its distance to reference coordinate $\mathbf{c}_r$.

We distinguish two cases, according to whether $\Pi < 1$ or $\Pi = 1$. The case where $\Pi < 1$ implies $\mathbf{c}_r \in \mathbf{A}_{\mathrm{MDP}}$ (Fig. 1.a), which follows from the definition of the migration recommendation predicate in (1) and the shape of $\mathbf{A}_{\mathrm{MDP}}$.

Clearly, nodes are more eligible when located closer to the reference coordinate. However, when the network is only sparsely populated, there may not be any nodes located close to the reference coordinate. In that case, the benefit of a migration will be small in comparison to what can be achieved for higher node densities. In such situations, it is more desirable to postpone migration to the next migration cycle until more suitable candidates become available. This is exactly the purpose of priority Π, which we apply to implement a deferral of a migration in such situations as follows.

Using Π, we are able to define a distance limit d_Π (Fig. 1.a) that any candidate node must satisfy as the maximum distance from $\mathbf{c}_r$. Nodes located farther from $\mathbf{c}_r$ will not be considered. For $\Pi < 1$, we define:

$$d_\Pi = 0.5 \cdot (1 + \Pi) \cdot d_{\mathrm{thresh}} \tag{3}$$

The interpretation of (3) is as follows: Consider $\Pi = 0$, which by definition leaves maximum freedom to the MDP to decide on a target node. In spatial terms, if $\mathbf{c}_r \in \mathbf{A}_{\mathrm{MDP}}$, a node must gain at least 0.5 times the distance d_{thresh} in order to be considered eligible. Choosing d_{thresh} instead of d_0 in (3) makes sure that progress is always relative to the threshold. This is important if u_0 is almost co-located with the reference coordinate, in which case no candidate nodes would exist in a very small region. For larger values of Π, distance d_Π increases, which effectively increases the set of nodes around the reference coordinate that will be considered in the selection process. The selection of the factor 0.5 is based on the fact that together with priority $\varPi$, it ensures that migration will only occur over a distance of at least r_{tx}.

In the second case, $\varPi = 1$, migration must occur according to the MRP's recommendation (which effectively turns into a decision in this case). Thus, we set $d_{\varPi} = d_{\mathrm{MDP}}$, where d_{MDP} is the maximum distance between the region and the reference coordinate (Fig. 1.b). That is, *any* node inside of $\mathbf{A}_{\mathrm{MDP}}$ is considered, even if it is located farther away from the reference coordinate than u_0 itself.

We are ready now to define the distance-based eligibility $\epsilon_{d,i}$ of each node u_i. Let $d_i(t_0) = d_i(t_i) + \bar{v}_i \cdot (t_0 - t_i)$, then

$$\epsilon_{d,i} = \begin{cases} 1 - d_i(t_0)/d_{\mathrm{thresh}} & \text{if} \quad \varPi < 1 \\ 1 - d_i(t_0)/d_{\mathrm{MDP}} & \text{if} \quad \varPi = 1 \end{cases} \tag{4}$$

In the previous equation we set $\epsilon_{d,i} = 0$ if $d_i(t_0) > d_{\mathrm{thresh}}$. The minimum distance-based eligibility, $\epsilon_{d,\varPi}$ is

$$\epsilon_{d,\varPi} = \begin{cases} 1 - d_{\varPi}/d_{\mathrm{thresh}} & \text{if} \quad \varPi < 1 \\ 0 & \text{if} \quad \varPi = 1 \end{cases} \tag{5}$$

The second portion of spatial eligibility is sojourn-based eligibility, which we denote by $\epsilon_{\Delta t,i}$. It describes the eligibility of a node to become a server in relation to the node's expected speed relative to the associated reference coordinate. The principal idea is that slow-moving nodes are to be preferred over fast-moving nodes, which is in particular beneficial for situations where stationary nodes are available. We omit further details on the derivation of the sojourn-based eligibility, which works very similar to the distance-based eligibility.

We will now combine the individual notions of eligibility into a single eligibility value ϵ_i for each node u_i. First, a sensible combination requires that the two individual eligibility values each obey their own limit. Second, compensation in one dimension may be allowed if the other dimension dominates, for example, when a fast-moving node is almost co-located with the reference coordinate.

We propose a multiplicative approach that we illustrate in Fig. 1.c. We have put the distance-based and sojourn-based eligibility magnitudes on the vertical and horizontal axis, respectively. The combined eligibility ϵ_i of a node and the minimum eligibility $\epsilon_{\varPi}$ that a node must obey are:

$$\epsilon_i = \epsilon_{d,i} \cdot \epsilon_{\Delta t,i} \quad \text{and} \quad \epsilon_{\varPi} = \epsilon_{d,\varPi} \cdot \epsilon_{\Delta t,\varPi} \tag{6}$$

Value $\epsilon_{\varPi}$ combines the requirements in each dimension into a one-dimensional value. In the case where a node u_i satisfies each individual eligibility, the overall limit is also satisfied, thus a node is considered eligible according to priority $\varPi$. For $\varPi = 1$, area $A_{\varPi}$ labelled ❹ in Fig. 1.c vanishes, thus any node is considered in the election process. In the figure, u_k clearly satisfies both the distance-based and sojourn-based minimum eligibility, thus it is eligible. Nodes u_i and u_j do not satisfy both individual eligibility values. However, u_i is able to compensate since the value defined by $\epsilon_{d,j} \cdot \epsilon_{\Delta t,j}$ is larger than $\epsilon_{d,\varPi} \cdot \epsilon_{\Delta t,\varPi}$.

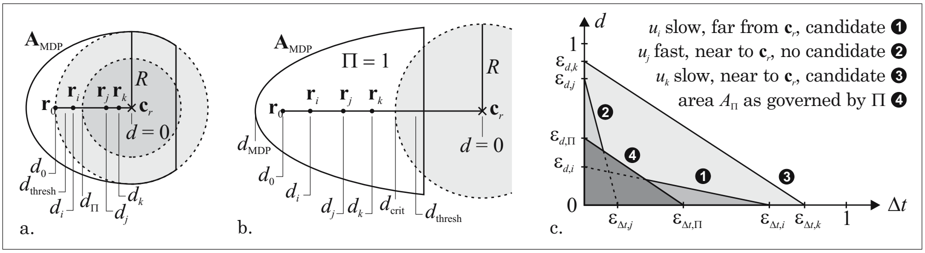

$\mathbf{A}_{\text{MDP}}$
$\mathbf{r}_0$
$\mathbf{r}_i$
$\mathbf{r}_j$
$\mathbf{r}_k$
$\mathbf{c}_r$
R
$d = 0$
d_0
d_{thresh}
d_i
d_Π
d_j
d_k
a.
$\Pi = 1$
d_{MDP}
d_{crit}
b.
d
1
$\varepsilon_{d,k}$
$\varepsilon_{d,j}$
$\varepsilon_{d,\Pi}$
$\varepsilon_{d,i}$
0
$\varepsilon_{\Delta t,j}$
$\varepsilon_{\Delta t,\Pi}$
$\varepsilon_{\Delta t,i}$
$\varepsilon_{\Delta t,k}$
Δt
u_i slow, far from $\mathbf{c}_r$, candidate ❶
u_j fast, near to $\mathbf{c}_r$, no candidate ❷
u_k slow, near to $\mathbf{c}_r$, candidate ❸
area A_Π as governed by Π ❹
c.

Fig. 1. a. and b.: Distance-based eligibility. — c. Node eligibility.

While (6) defines the eligibility of each considered node, it does not capture whether a node is connected to the current server u_0 over a sufficiently stable network topology. For that purpose, we incorporate the notion of path stability based on the estimated residual path lifetime. Considerable work has been done in the field of link and path stability, e.g. in [5], and we will omit the details of how suitable paths are determined. For ease of discussion, we assume that P_T is a sufficiently stable initial path returned by the migration decision policy, computed based on the information returned during state exploration. We note that, depending on the amount of data to be transferred and to account for the dynamics in the network topology, successive state explorations and path computations are performed to update P_T whenever necessary.

We now define predicate P_{MDP} that finally decides on a migration:

$$P_{\text{MDP}} \quad := \quad \exists i : \epsilon_i \geq \epsilon_{\Pi} \quad \wedge \quad \exists P_T \tag{7}$$

In other words, migration to some node u_i will occur if a node exists that has at least the eligibility defined based on priority Π and a target path, i.e. a sufficiently stable initial path exists to that node. Observe that an eligible node may not be reachable over a sufficiently stable path or vice versa.

To determine a target node, we will make use of the auxiliary set $I = \{i \mid \exists \text{ sufficiently stable path to } u_i\}$. Then, the target node u_T is

$$u_T \quad := \quad u_i : i \in I \quad \wedge \quad \forall j \in I : \epsilon_i \geq \epsilon_j \tag{8}$$

In this form, the target node with the largest eligibility is selected. If the migration decision predicate (7) is false, the policy is deferred to the next monitoring cycle and the migration recommendation policy is evaluated again. Note that we have omitted the fact that the migration decision policy may trigger repetitively. This can be avoided by introducing a blocking interval after each evaluation of a migration decision policy to avoid frequent state explorations.

5 Evaluation

In this section we assess the performance of the migration policies described in the previous chapter based on an implementation in the ns-2 simulator. For that purpose, we have extended the storage approach in [3] with a migration protocol that fully implements the policies proposed in this paper. We will refer to this migration protocol by DataMiP in the remainder of this section. In principal, any other approach that builds on the server abstraction and geometric routing can be used as a substrate for our migration policies.

The migration of a data subset is initiated by the current server u_0 after the migration decision predicate in (7) has become true. According to the migration decision policy, the migration process takes target server u_T from (8) and path P_T as input. DataMiP utilizes a reliable transmission protocol based on source routing for transferring a data subset between u_0 and u_T. We will discuss geometric routing as an alternative in the experimental results. Sliding window-based flow control tuned to wireless multihop networks using window sizes according to [2] is used to achieve optimum throughput. If the current estimate of a path's lifetime is too small to transfer the data subset under the given node mobility, DataMiP performs on-the-fly path switching according to the migration decision policy. Recomputing a new path sufficiently in advance before the current path's lifetime elapses provides virtually full performance transparency, as long as such a path can be determined.

Due to the lack of comparable approaches in the literature, we have implemented two additional straightforward policies that we will use for performance comparison: greedy and progressive migration. In *greedy migration*, upon reaching threshold distance d_{thresh}, we use greedy routing to find the nearest greedily reachable node to the reference coordinate. Then, GPSR is used for the data transfer to that node. The second approach, which we call *progressive migration*, is initiated upon reaching d_{thresh} as well. Then, from the known neighbors to the current server, the closest one to the reference coordinate is selected as the migration target. Once a single migration in the progressive approach has completed, neighbor search and additional migrations are repeated until no node closer to the reference coordinate can be found.

The parameter settings of our simulations are as follows. We assume a fixed simulation area of 600×600 m^2 and a single reference coordinate located at its center. The transmission range of nodes is fixed at 100 m. The default values for the number of nodes is 150, and a default migratable data size of 320 kByte is assumed. The default threshold distance d_{thresh} is 100 m. We use the random waypoint mobility model, where nodes are partitioned according to their speed. We assume a speed mix ratio of 1:5, that is, 20 percent of the nodes move at 2 m/s, while the rest moves at 10 m/s. The pause time is always 30 s. We have averaged over 5 simulation runs, each with a duration of 3600 seconds.

We consider the following performance metrics. *Spatial coherence* quantifies the ability of each migration policy to maintain a certain level of proximity between a data subset (that is, the active server) and the reference coordinate. It is defined as the mean geometric distance of the current server from the reference coordinate over time. *Migration efficiency* describes the communication costs that are required to maintain a certain level of spatial coherence. It is defined in units of mean aggregated packet size per unit time. It includes the data contained in all packets that are part of the migration approach, in particular, the MDP's state exploration packets and all data that is subject to migration. *Migration latency* defines the percentage of the aggregated time migrations are in progress. Finally, *migration stability* is the essential metric that describes the success ratio of migrations, which is to be maximized. We express migration stability in terms of the mean number of migration failures per unit time and show its relation to the mean number of migrations that occur in a scenario.

Fig. 2.a depicts our results for spatial coherence as a function of the migratable data size. DataMiP is shown in two configurations. DataMiP/SRT refers to the case where source routing via path P_T as determined by the MDP is used for migration. In DataMiP/BPR data is transmitted greedily to the target node determined by the MDP. Note that this is very different from greedy migration, which does not implement any advanced policy, let alone stability analysis. The results in Fig. 2.a indicate that for any considered data size of 320 kByte and below, all of the four approaches perform in a similar way. In the region above about 320 kByte, both DataMiP configurations perform significantly better. The reason for the degradation of both greedy and progressive migration is that they do not differentiate between slow and fast nodes when selecting a target node, other than the MDP. Therefore, with increasing data size, fast target nodes selected more often by greedy and progressive migration than by DataMiP will move away from the reference coordinate more quickly than slow nodes favored by DataMiP's migration decision policy.

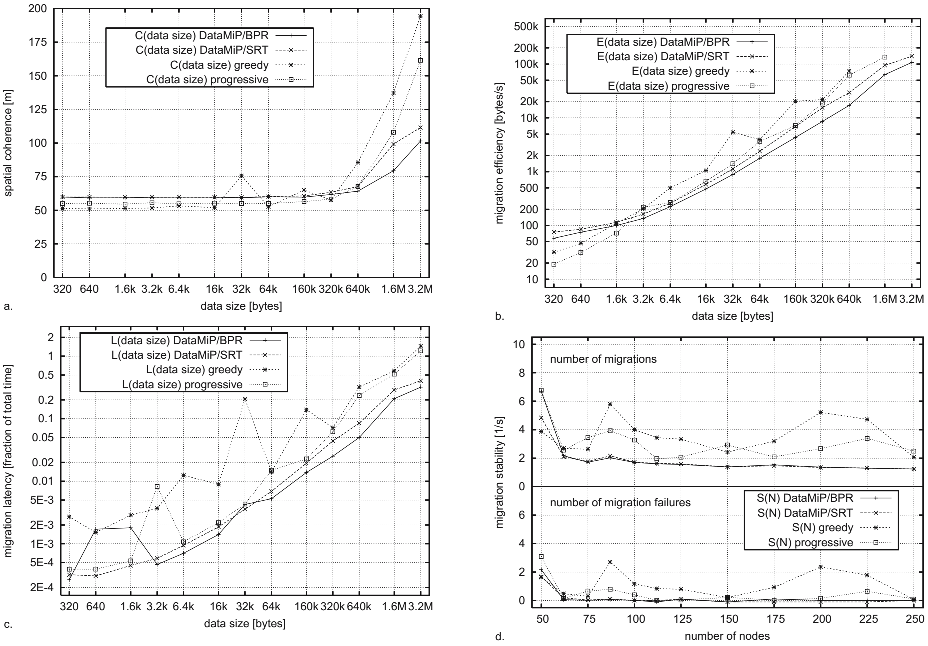


Fig. 2. Evaluation: Simulation Results

Fig. 2.b shows migration efficiency as a function of the the data size. One observation is that both configurations of DataMiP generally have better performance than greedy and progressive migration, except for the lowest range in the diagram. Greedy and progressive migration have a true advantage for a data size of about 1.6 kByte and below. The reason is that the migratable data size is so small that a single packet is sufficient to transfer the whole data. In that case, the MDP's state exploration requires many packets in relation to the transferred data packets. Clearly, for this special case of extremely small data sizes, no particular effort is required for migration. On the other hand, DataMiP performs significantly better above 64 kByte. Note that the vertical axis is logarithmic scale, and performance varies significantly between DataMiP and the straightforward migration approaches.

Results for the migration latency are given in Fig. 2.c as a function of the data size. Besides the general performance advantage of both configurations of DataMiP, in particular, DataMiP/BPR, the smoothness of the two curves in the graphs shows extremely well predictability of latencies. In contrast, both greedy and progressive migration exhibit a very erratic behavior. Observe that for 3.2 MB, migration latency is above a theoretical maximum of 60 min for both greedy and progressive migration. This is due to the fact that migration failures in these approaches lead to redundant servers, whose migration delay we have aggregated to point out this issue. Finally, Fig. 2.d underlines DataMiP's robustness, shown by example as a function of the number of nodes. Observe that for all settings, DataMiP/SRT performs best, indicated by the fact that there are virtually no migration failures (bottom graph), except for an extremely low node density of just 50 nodes per considered area. The high performance can be attributed to the benefit of the MDP's stability analysis.

6 Conclusion

This paper presented novel policies for data migration to support location-centric data storage in mobile ad-hoc networks. Strategies were defined that enforce spatial coherence between data and an associated geometric reference coordinate. The proposed policies were implemented within a migration protocol and integrated with an existing location-centric storage approach.

Experimental results indicate that the proposed migration policies provide superior performance over more straightforward approaches with respect to the considered performance metrics. While supporting large quantities of migratable data, highly mobile nodes, and sparse network densities, spatial coherence is effectively maintained at low communication costs and migration latency. Stability results demonstrate the robustness the policies provide in conjunction with a reliable transmission protocol and failures are virtually eliminated for most parameter settings in contrast to the straightforward approaches.

More qualitative aspects include the ability of the migration framework to be used independently of any geometric routing protocols, since migration can rely on source routing only. This allows the rest of the network to be optimized individually with respect to geometric routing without impacting migration performance. Furthermore, the policies have significant potential to extensibility. While we have focused only on the most relevant aspects of node eligibility that build on spatial characteristics, additional notions, e.g. load balancing or energy issues can easily be accommodated. For that, state exploration can be extended to return additional information from remote nodes.

References

1. Araújo, F., Rodrigues, L., Kaiser, J., Liu, C., Mitidieri, C.: CHR: A distributed hash table for wireless ad hoc networks. In: ICDCSW 2005. Proc. 25th IEEE Int'l Conf. Distr. Comp. Syst. Workshops, Columbus, OH, USA, pp. 407–413 (June 2005)
2. Chen, K., Xue, Y., Shah, S.H., Nahrstedt, K.: Understanding bandwidth-delay product in mobile ad hoc networks. Comp. Comm. 27(10), 923–934 (2004)
3. Dudkowski, D., Marrón, P.J., Rothermel, K.: Efficient algorithms for probabilistic spatial queries in mobile ad hoc networks. In: COMSWARE 2006. Proc. 1st Int'l Conf. Comm. Syst. Software and Middleware, New Delhi, India (January 2006)
4. Dudkowski, D., Marrón, P.J., Rothermel, K.: An efficient resilience mechanism for data centric storage in mobile ad noc networks. In: MDM 2006. Proc. 7th Int'l Conf. on Mobile Data Management, Nara, Japan (May 2006)
5. Gerharz, M., de Waal, C., Frank, M., Martini, P.: Link stability in mobile wireless ad hoc networks. In: LCN 2002. Proc. 27th Ann. IEEE Conf. on Local Comp. Networks, Tampa, FL, USA, pp. 30–39 (November 2002)
6. Ghose, A., Grossklags, J., Chuang, J.: Resilient data-centric storage in wireless adhoc sensor networks. In: Chen, M.-S., Chrysanthis, P.K., Sloman, M., Zaslavsky, A. (eds.) MDM 2003. LNCS, vol. 2574, pp. 45–62. Springer, Heidelberg (2003)
7. Karp, B., Kung, H.T.: GPSR: Greedy perimeter stateless routing for wireless networks. In: MobiCom 2000. Proc. 6th Ann. Int'l Conf. Mobile Comp. and Networking, Boston, MA, USA, pp. 243–254 (August 2000)
8. Kieß, W., Füßler, H., Widmer, J., Mauve, M.: Hierarchical location service for mobile ad-hoc networks. Mobile Comp. and Comm. Rev. 1(2), 47–58 (2004)
9. Landsiegel, O., Götz, S., Wehrle, K.: Towards scalable mobility in distributed hash tables. In: P2P 2006. Proc. 6th IEEE Int'l Conf. Peer-to-Peer Comp, Cambridge, UK, pp. 203–209 (September 2006)
10. Rao, A., Papadimitriou, C., Shenker, S., Stoica, I.: Geographic routing without location information. In: MobiCom 2003. Proc. 9th Ann. Int'l Conf. Mobile Comp. and Networking, San Diego, California, USA, pp. 96–108 (September 2003)
11. Ratnasamy, S., Karp, B., Shenker, S., Estrin, D., Govindan, R., Yin, L., Yu, F.: Data-centric storage in sensornets with GHT, a geographic hash table. Mobile Networks and Applications 8(4), 427–442 (2003)
12. Seada, K., Helmy, A.: Rendezvous regions: A scalable architecture for service location and data-centric storage in large-scale wireless networks. In: IPDPS 2004. Proc. 18th Int'l Parallel and Distr. Processing Symp, p. 218 (2004)
13. Tei, K., Sommer, C., Fukazawa, Y., Honiden, S., Garoche, P.-L.: Adaptive Geographically Bound Mobile Agents. In: Cao, J., Stojmenovic, I., Jia, X., Das, S.K. (eds.) MSN 2006. LNCS, vol. 4325, Springer, Heidelberg (2006)
14. Wu, X.: VPDS: Virtual home region based distributed position service in mobile ad hoc networks. In: ICDCS 2005. Proc. 25th IEEE Int'l Conf. Distr. Comp. Syst, Columbus, Ohio, USA, pp. 113–122 (June 2005)

An Efficient and Low-Latency MAC Protocol for Wireless Sensor Network

Zhichao Gu and Jifeng Sun

School of Electronics and Information,
South China University of Technology,
Guangzhou, China, 510640
gzhichao@gmail.com, ecjfsun@scut.edu.cn

Abstract: This paper proposes EL-MAC, a contention-based medium access control (MAC) protocol, which is efficient and low-latency for the wireless sensor network (WSN). EL-MAC introduces duty-cycle and virtual cluster scheme within the framework of S-MAC to reduce energy consumption and to self-organize network. Besides, Inspired by D-MAC, the scheme of data forwarding chain (DFC) is proposed for reducing the latency in multi-hop transmission. The experiment of simulation shows that EL-MAC has lower latency and higher throughput with comparative energy consumption on different traffic load condition than S-MAC.

1 Introduction

Wireless sensor networking (WSN) is one of the latest technologies and has been rapidly developed recently. It has a wide range of potential applications in many areas, such as environment detection, target tracking, industrial control and medical care, etc. Normally, WSN consists of hundreds or thousands of micro sensor nodes that organize themselves as wireless multi-hop network. Compare with traditional wireless networks, WSN is more constrained by limited energy resource, low storage capacity and restricted communication bandwidth. Therefore, Protocols are necessary to be reconsidered, instance topology, routing, MAC and synchronization.

This paper makes the study on the medium access control protocol of WSN. Here not only energy conservation is focused on as most previous researchers do, but also the efficiency should be considered. S-MAC [2] is an efficient protocol for WSN and conserves energy by using duty-cycle, overhearing avoidance and message passing schemes. However, strict schedulers of S-MAC incur *data forwarding interruption* (DFI) problem in multi-hop transmission since packets must wait for next cycle to be forwarded. The DFI problem brings on high latency, and moreover, the delay time accumulates hop by hop and is intolerable for real-time system. For solving the DFI problem, the author of S-MAC has appended adaptive listening scheme [3] into his design, yet it can just forward one more hop in a cycle. Another method is proposed in D-MAC [7], which is designed by using *data gathering trees* to collect data from several sources to one sink to reduce latency. Nevertheless, its TDMA-Based scheme requires high-precision synchronization and be less expansible in dynamic network.

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 209–220, 2007.

Inspired by D-MAC, EL-MAC utilizes *data forwarding chain* (DFC) to transfer packets hop by hop. Dissimilarly, the *data forwarding chain* is built by contending means. According to the DFC scheme, correlative nodes staggered wake up in proper time so that packets could be duly forwarded as more hops as possible in a cycle. Forwarded hops of a packet in a DFC not only relate to packet's size and periodic time, but also relate to network traffic and contention situation, which will be detailed in the later text.

The experiment environment of simulation for our protocol is described and the performance results shows that EL-MAC has lower latency and higher throughput with comparative energy consumption on different traffic load condition than S-MAC. Drawbacks and future work are also presented in the lattermost part of the paper.

2 Related Work

Mass research work has been done for MAC protocol of sensor network. Previous MAC protocols can be classified as contention-based and TDMA class basically. S-MAC [2] [3] is contention-based and the most relevant protocol for our approach. It introduces schemes from standardized IEEE 802.11 distributed coordinate function [1], and proposes an active/sleep cycle scheme to conserve energy. But its latency is high in multi-hop network. T-MAC [4] is developed from S-MAC, but decreases energy waste of idle listening. However, it doesn't improve throughput from S-MAC and introduce another early-sleep problem. Sift [5] is a novel contention-based protocol. Different from others, Sift choose probability in a fixed-sized contention window to send. But it need precise synchronization between nodes and not suitable for flat routing network. TDMA-Based MAC [6] is designed for cluster network, whose cluster heads assign time-slots for normal nodes. But this protocol requires extra appropriate cluster-head chosen algorithm and may consume more energy. D-MAC [7] is another relevant protocol for our design. It is also TDMA-based and has low latency by utilizing data gathering trees. However, it only suits cluster network, for data is gathered from multi-source to one sink. Besides it requires higher precise synchronization that increases the complexity of synchronized scheme. A-MAC [10] tunes its work/sleep status adaptively determined base on network traffic. The energy is efficient especially in light traffic.

3 EL-MAC Protocol Design

EL-MAC is a contention-based MAC protocol with periodic cycle and virtual cluster scheme. Although energy conservation is the first factor in MAC design for WSN, the latency problem needs much regarding. EL-MAC utilizes DFC to decrease network latency yet its energy consumption is comparative with S-MAC.

3.1 Basic Scheme

EL-MAC introduces duty-cycle scheme of S-MAC, its basic scheme is shown in Fig. 1. Each node is active and asleep periodically. As duty-cycle scheme of S-MAC, synchronized (SYNC) message is periodically broadcasted in the "for SYNC" phase, which is arranged as prophase of active time. The SYNC message is used to exchange schedulers with immediate neighbors and remedy their clock drift. Normally, the period for sending SYNC message is about several times more than the cycle period.

In the "for RTS" phase, when a node has data to transmit, it sends RTS (request to send) message if only it has won the medium. Then nodes that are idle just listen in the medium, and prepare for receiving or forwarding. Basically, our protocol follows the sequence as RTS/CTS/DATA/ACK for transmission, yet with a little difference. This part will be discussed detailedly in next section.

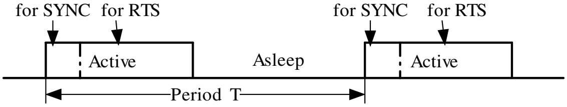


Fig. 1. Basic scheme

3.2 Data Forwarding Chain (DFC)

In our protocol, a uni-cast packet is delivered covering multi-hop through the *data forwarding chain* (DFC), which is set up temporarily and dynamically. The lifetime of a DFC is not longer than a cycle time. Since nodes with common virtual cluster wake up and sleep simultaneously, we divide the cycle time into *sync period*, *listen period* and *sleep period*. As names imply, the first period is only used to broadcast SYNC message while the second one is used to deliver RTS or CTS message specially. However, the *sleep period* is used for either sleeping if at leisure time or transmitting DATA packet otherwise.

Therefore as to correlative nodes, the process of transferring packets with DFC mainly go through following steps, which are: 1) request/reply to set up a DFC within *listen period*; 2) sleep in *Wait-Forward* state to conserve energy; 3) staggered wake up to send/receive DATA packet within *sleep period*; 4) logout the DFC and sleep as soon as sending out the DATA packet. Fig. 2 illustrates the packet transmission of 4 nodes with *data forwarding chain*.

For example, when a node intends to transmit DATA packet, it contends the medium firstly and then sends out RTS within *listen period*. Its neighbor receives this request message and continues to forward it to next-hop after another contention. The RTS message contains node's rank number and act as both request for next-hop and reply for prior-hop. So the sender will receive its neighbor's RTS and know that their connection has been established. Then the sender switches into *Wait-Forward* state and goes into sleep. The forwarder, who is the sender's neighbor mentioned before, is now in *Wait-CTS* state and won't goes to sleep until it receives confirm reply (RTS or CTS) from it's next hop. By such and such, request message is finally forwarded to the receiver hop by hop. The receiver sends a CTS message back to its prior-hop as soon as receiving the RTS message. Therefore So far, a DFC from the sender to the receiver has been set up. Each correlative node of DFC switches into *Wait-Forward* state and sleeps when it receives confirm reply from its next-hop. The sleeping time is reserved for each node according to the size of DATA packet. When each node's timer times out, it wakes up to transmit DATA.

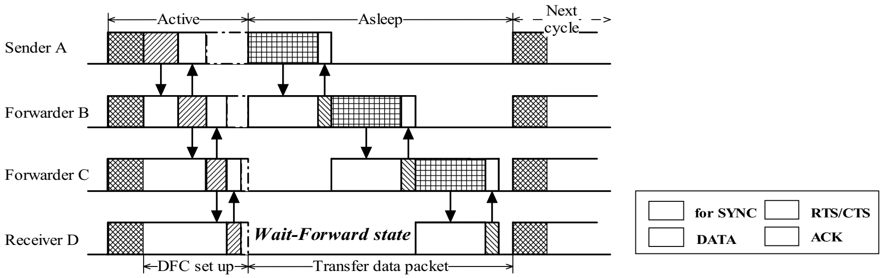


Fig. 2. Data Forwarding Chain (DFC)

As Fig. 2 shows, correlative nodes staggered wake up within *sleep period* to send or receive DATA packet, the forwarding mechanism is similar to *data gathering tree* of D-MAC in some degree.

3.3 Contention Mechanism

Since our protocol utilizes the contention-based mechanism to make network more flexible and expansible, traditional carrier sense and back-off method become improper for setting up a DFC. In order to increase the probability of building the forwarding chain, a special contention mechanism is designed for EL-MAC.

As for a node, the length of its contention window CW_i is related to the node's rank number '*i*' in DFC. The larger the node's rank is, the smaller its contention window should be. Then CW_i is calculated with following equation:

$$CW_i = \begin{cases} \dfrac{CW_{\max}}{2^i}; & i = 0,1,2... \\ CW_{\min}; & if\,(CW_i \le CW_{\min}) \end{cases} . \tag{1}$$

where CW$_{max}$ and CW$_{min}$ are fixed preset value. Then, by calculating the modulus of a random value with CW_i, the time for carrier sense t_{cw} is calculated as following function, where $t_{timeslot}$ is the time slot unit.

$$t_{cw} = [random() / CW_i]_{MOD} \times t_{slottime} \ . \tag{2}$$

By this means, since contention window is smaller, a node in rear position of a DFC may still have much chance to win the medium. In other words, the longer a DFC is established, the larger probability it can continue to add next hop into itself.

For those who fail in medium contention, there are two situations for discussion. On the one hand, if the node is of first rank (sender) to send RTS, it suspends present request and goes into sleep after renewing its *network allocate vector* (NAV) value. The NAV value records the remaining time of medium being taken and be decreasing all the time. When it clears to zero or next cycle arrives, the node wakes up and retries contending again. However, DATA packet that needs to be sent will be dropped if the node has retried more than a certain times.

On the other hand, if the node is a forwarder, it suspends present request and not retries contending again. Then its prior-hop will time out of waiting confirm message and consider itself as the last hop of the established DFC. Thus the only effect is that the forwarding chain's size won't reach the max-hops that it supposed to be.

3.4 Size Evaluation of Data Forwarding Chain

Now the question is how many hops a packet can be transferred in a cycle time on ideal condition, which means there is no failure or interruption happens when building DFC and transmitting. The answer is direct proportion to the ideal size of DFC.

As shown in Fig. 2, the *listen period* can be logically divided into several sending timeslots of different length. Each timeslot represents the max time for transmitting a RTS/CTS message of different rank, including the time for contention. Therefore the sending timeslot mainly consists of time for contention window and control message duration. As for the i^{th} node in a DFC, the structure of sending timeslot is illustrated in Fig. 3 and its value should be calculated as:

$$t_{\alpha}^{i} = (CW_i \times t_{slottime}) + t_{ctrl} + t_{difs} \ . \tag{3}$$

where t_{ctrl} represents time for control message (RTS/CTS) duration, t_{difs} represents interval time between message frames.

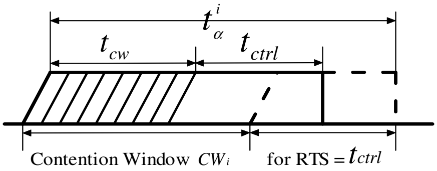


Fig. 3. Structure of Sending Timeslot of rank = *i*

Furthermore, Fig. 2 shows that the time for transfer DATA and ACK is changeless. It's considered as an integrated unit which consists of time for DATA packet duration and control message duration and an interval time as well. It is detailed as follows:

$$t_{\beta} = t_{data} + t_{ctrl} + t_{difs} \ . \tag{4}$$

where t_{data} represents time for DATA packet duration.

Since the cycle time is fixed, the ideal size of a DFC is therefore related to duty cycle and the length of DATA packet. Supposed *N* is the ideal size of a DFC, then the max hops included by the DFC is *N*, it should obey functions as follows.

$$\sum_{i=0}^{N} t_{\alpha}^{i} \le t_{listen} \tag{5}$$

$$N \times (t_{\beta} + T_{wait}) \le t_{sleep} \tag{6}$$

where t_{listen} and t_{sleep} represent time of *listen period* and *sleep period* respectively. Besides, notice that neighbor nodes can hardly wake up exactly at the same time due to clock drift, sending DATA immediately after time out of *Wait-Forward* timer may result in error. So waiting a random short time is helpful for solving this problem. Wherefore T_{wait} represents max time for a sender/forwarder to wait.

According function (5), RTS/CTS messages can only be sent within *listen period* for requesting or replying, the ideal size of a DFC is direct ratio to the time of *listen period*. Again, function (6) illustrates that the transmission of DATA with DFC should start immediately when *sleep period* arrives, and should finish before ending of the present cycle. Since the time of *sleep period* is fixed, *N* is inverse ratio to the DATA packet size.

3.5 Timers and Interruption Avoidance

As mentioned before, forwarded hops of a packet not only relate to the ideal size of DFC, but also relate to network traffic and contention situation. Since EL-MAC is contention based mechanism, forwarding process may be interrupted due to several reasons, which could be concluded as follows:

1) Failure of requesting or replying for DFC within *listen period*.
2) Some nodes couldn't wake up from *Wait-Forward* state at proper time.
3) Errors that happens when DATA packet is being transferred.
4) Collision from some other nodes which is not included by the present DFC.
5) Early sleep of nodes in DFC.

Wherefore, nodes maintain several special timers for solving the prior three reasons. Moreover, collision avoidance and early sleep avoidance mechanisms are utilized in our protocol for solving reason 4) and 5).

3.5.1 Wait-CTS Timer

This timer is initialized with the max time for waiting confirm reply from next hop immediately when RTS message is sent out. If receive confirm message (RTS/CTS) within period of timing, a node will cancel *Wait-CTS* timer and switch into *Wait-Forward* state and sleep. Otherwise, if receive no reply until the overtime, this node will retry to sent the RTS again unless the left time of current *listen period* is un-enough for initializing the *Wait-CTS* timer. Notice that all RTS requesting should be over at the end of *listen period* since DATA transmission starts when *sleep period* arrives. Therefore a node will be considered as the final hop of present DFC if it receives no reply message eventually. Then this node drops current RTS request and goes to sleep with state being switched to *Wait-Forward* state until it's waken up by the *Wait-Forward* timer.

Obviously, initial values of the timer for each node in DFC are different. As for the i^{th} node in DFC, the initial value equal to the sending timeslot t_{α}^{i+1}, which implies the max time for its next-hop node to sent a reply message out.

3.5.2 Wait-Forward Timer

This timer is used for recording the period of sleeping time in *Wait-Forward* state, it is initialized when a node receives the confirm reply in the process of RTS requesting.

The node will be wakened by this timer when it's time out. Then if the node is a sender whose rank number is zero, it will send out the DATA packet after waiting for a random short time. Otherwise, if the node is a forwarder or receiver, it will switch into *Wait-DATA* state and be ready to receive the DATA packet.

Since the transmission of DATA packet in DFC begins when *sleep period* arrives, the initial value reserved for this timer is calculated as following function:

$$t^{i}_{WaitForward} = \begin{cases} (t_{listen} - T_{now}) + (i-1) \times t_{\beta}; & i > 0 \\ (t_{listen} - T_{now}); & i = 0 \end{cases} \tag{7}$$

where '*i*' is the rank number of the node in DFC, and T_{now} is the present time of initializing the *Wait-Forward* timer. Obviously the second equation implies remaining time of current *listen period* while the latter part of first equation denotes the ideal duration for DATA be transferred from the sender to the current node.

3.5.3 Wait-DATA Timer

This timer is initialized with a max time for receiving DATA packet from prior hop immediately when a forwarder or receiver is wakened by the *Wait-Forward* timer. Then the node switches into *Wait-DATA* state until DATA has been received or the *Wait-DATA* timer fires. If receives the DATA packet within the period of timing, the node sends back an ACK message at once, and then it forwards the DATA packet to its next hop after a short random time unless it's the final hop of DFC. Otherwise, if no DATA be received when the timer is time out, the node stops waiting and sleeps until next wake up phase arrives.

Now the question is how to evaluate the time for initializing the timer. Notice that a short random waiting time is necessary before sending out DATA for each node in DFC. However, the time of *Wait-Forward* timer fires is preset on the ideal condition that there is no delay when DATA is being transferred. Hence this short waiting time should be considered when evaluating the max time for waiting DATA packet. Since the random time is accumulated hop by hop while its max value is T_{wait} for each hop, then the period of time for initializing the *Wait-DATA* timer could be calculated as:

$$t^{i}_{WaitData} = i \times T_{wait} + t_{data} \ . \tag{8}$$

where '*i*' is the rank number of the node.

3.5.4 Wait-ACK Timer

This timer is scheduled for receiving ACK reply message and be initialized immediately DATA packet is sent out. Obviously its initial scheduled value at lease is the duration for transmitting an ACK message, which is equal to t_{ctrl}. If receives ACK from next-hop within timing period, the node will release the memory for DATA packet that just be sent, and sleep until next wake up phase. Otherwise, if no ACK be received when the *Wait-ACK* timer fires, the node will retry to send the DATA packet again unless its retry times have reached a preset max times. In this case, current transferring will be abandoned, and the DATA packet will be handed up to the upper routing layer. Therefore when next cycle arrives, this node will contend to send RTS again and act as a sender. New DFC is supposes to be set up so that the DATA packet could be transferred sequentially in this new cycle.

3.5.5 Collision Avoidance Mechanism

In order to avoid collision incurred by outer nodes of DFC, similar mechanism to S-MAC is utilized in our protocol. As mentioned before that each node maintains a network allocate vector (NAV), which records the remaining time of medium being taken and decreases all the time. Besides, every message frame contains a value of reserving time of medium, which usually last out until DATA transmission has finished. Therefore if overhears a massage, a node will renew its NAV with this value and sleep to avoid collision with others until its NAV is clear. Since the lift time of a DFC is limited in current cycle, the time of medium being taken is not more than a cycle period.

3.5.6 Early Sleep Avoidance Mechanism

Early sleep will result in the forwarding interruption of DFC. The first reason for early sleep is collision avoidance. As Fig. 4 shows, if a node receives a RTS message that destines to others, it is supposed to go to sleep to avoid collision. However, this node may be a potential member of the current DFC, thus its sleepiness will block the latish request for extending the chain.

In this case that a node couldn't predict if it's a potential member when it overhears a RTS message, it will keep awake and listening for a certain period of t_{α}^{i+1}, which is the max time to receive RTS. Here '*i*' is the rank number of the node who has sent the overheard RTS message. Therefore, if it receives no requesting message within this period or the received message is destine to others, the node will sleep for a scheduled time of NAV. Otherwise, it should reply or forward the received request.

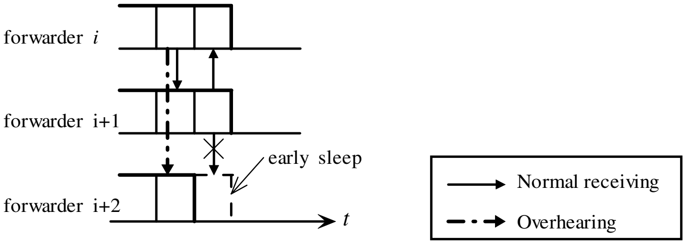


Fig. 4. Early sleep caused by overhearing

Another reason for early sleep is timeouts of *Wait-DATA* timer. For example, a node is supposed to go to sleep if no DATA packet is received when the *Wait-DATA* timer fires. However, this node may have overheard the ACK message from prior hop node within *Wait-DATA* state. In this case, since DATA will be forwarded immediately after ACK being sent, the node's *Wait-DATA* timer should be rescheduled and the waiting time should be prolonged for a certain period of t_{data}, so that the latish DATA packet could be received successfully. This process is illustrated in Fig. 5.

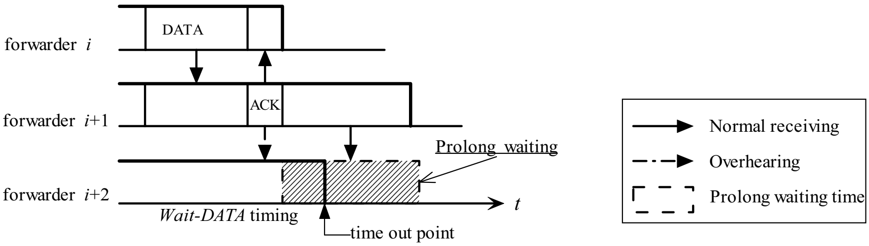


Fig. 5. Early sleep caused by timeouts of Wait-DATA timer

4 Performance Evaluation

Our protocol is implemented in the NS-2 network simulator for experiment. Its performance is compared with S-MAC under an assumed environment, which will be described in subsection.

4.1 Assumed Experiment Environment

As Fig. 6 shows, a simple topology is used for our experiment. There are 6 nodes in topology with transmitted range of 100m. The space distance among each other is 90m, which means only communication of immediate neighbors is allowed. The *cbr0* packets from source 1 flow through node 2 and 3 and end at sink 4, while the *cbr1* packets from source 5 through node 3 but end at sink 6.

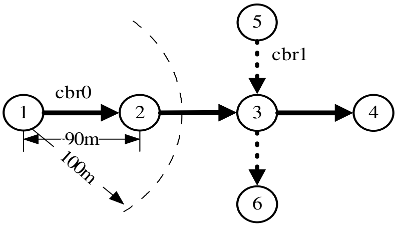


Fig. 6. Topology of the experiment

The simulation lasts for 2000s. In the first 1000s only *cbr0* flow is generated. Its packet inter-arrival period is changed to simulate traffic load variation of network. The less the inter-arrival period is, the heavier the traffic load will be. As from 1000s to 2000s, *cbr0* flow keeps its packet inter-arrival period as 2s. The *cbr1* flow starts from 1200s and stops at 1500s with packet inter-arrival period of 3s.

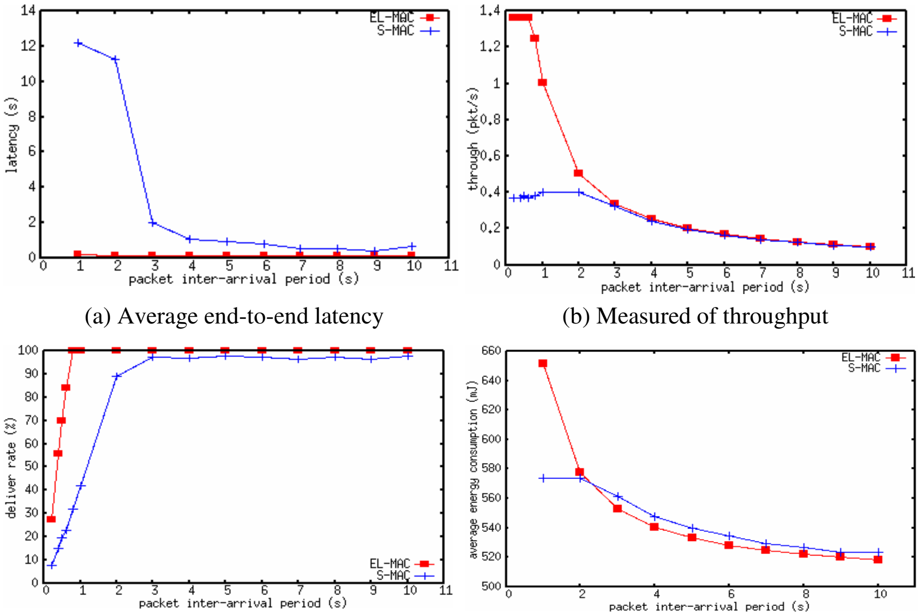
4.2 Performance Results

The experiment is compared between S-MAC and EL-MAC, of which both are set to have same cycle period and duty-cycle. Besides, they both work together with DSR routing protocol in the simulation.

Fig. 7 shows the comparison of performance results when there is only *cbr0* flow in the network, where x-coordinate denotes the packet inter-arrival period. Obviously the network traffic load is heavy when the inter-arrival period is more than 3s, and the traffic load is light when the x-coordinate value is less than 3s contrarily.

As shown in Fig. 7(a), the average end-to-end latency of EL-MAC keeps a steady low level all through and be much less than that of S-MAC. This result is especially obvious when the traffic load is heavy.

Fig. 7(b) shows the measured throughput of EL-MAC and S-MAC. According to the DFC scheme, packet is forwarded more hops in a cycle with EL-MAC, so more packets could be transmitted in a unit time. Therefore the throughput of EL-MAC is much higher than that of S-MAC. However, the throughput of both MAC descends smoothly to be nearly the same when fewer packets are generated in light traffic load.



(a) Average end-to-end latency

(b) Measured of throughput

(c) Measured of delivery rate

(d) Measured of energy consumption

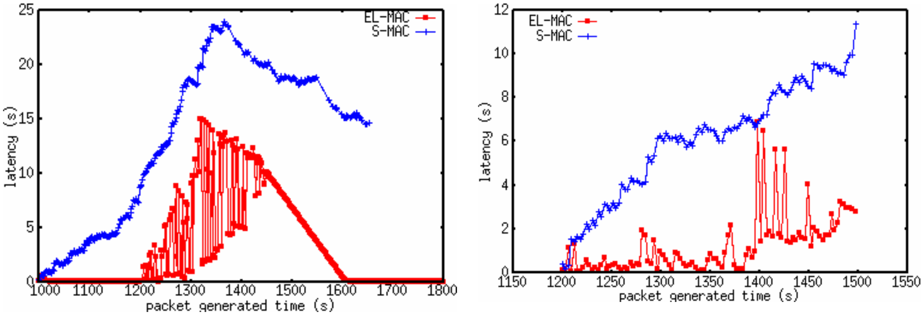
Fig. 7. Comparison of Performance results

Fig. 7(c) shows that the delivery rate of both MAC protocol is high when there is little traffic in network. However when the traffic load becomes heavy, both protocols descend quickly. Yet EL-MAC has a higher delivery rate than S-MAC. Notice that packets could be forwarded betimes with DFC in our protocol, there are few dropped packets incurred by memory overflowing.

Fig. 7(d) shows that the average consumption of EL-MAC is slightly less than that of S-MAC in normal traffic load. The reason why EL-MAC consumes more energy than that of S-MAC in heavy traffic is because our protocol has to consume more to maintain a higher delivery rate. Therefore, the consumption of transmitting a packet is less with EL-MAC than that with S-MAC.

Fig. 8 shows the comparison of efficiency in contended environment. When *cbr1* starts at 1200s, packets of *cbr0* and *cbr1* flow at one time. The traffic is extremely heavy in network while contention and collision are supposed to happen. Therefore, the latency of EL-MAC and S-MAC both increase quickly. However, our protocol performs better for both *cbr0* and *cbr1* flow.

Besides, it's interesting that the curve of EL-MAC vibrates acutely in Fig. 8 for both *cbr0* and *cbr1* flow. This is due to the interruption avoidance mechanism. As a DATA packet is interrupted from being transmitted in a DFC, it would be sent with another new DFC in next cycle. More cycles have to be expended for transmitting this DATA packet. Even so, efficient of EL-MAC is no worse than S-MAC on the worst condition.



(a) Latency of *cbr0*

(b) Latency of *cbr1*

Fig. 8. Measured latency in contended environment

5 Conclusion and Future Work

This paper has proposes a new MAC protocol for wireless sensor network. Compare with S-MAC, our protocol has higher efficiency in latency and throughput, yet its energy consumption is comparative with S-MAC. The protocol has been implemented in NS-2 simulator and has achieved a good performance.

However, there are some drawbacks that need to be amended. Firstly, nodes in *data forwarding chain* must know a whole routing information. In experiment we utilize the DSR routing protocol which is not quite efficient for WSN. Secondly, In order to request multiple hops for connection, EL-MAC has to listen for more time than S-MAC in a cycle.

For solving these drawbacks, in future we are going to develop a better routing protocol. Moreover, traffic-based adaptive duty-cycle mode can be appended into the protocol for conserving more energy.

References

1. LAN MAN Standards Committee of the IEEE Computer Society. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, IEEE Std. 802.11-1999
2. Ye, W., Heidemann, J., Estrin, D.: An Energy-Efficient MAC Protocol for Wireless Sensor Networks. In: INFOCOM 2002. Proceeding of the 21st International Annual Joint Conference of the IEEE Computer and Communications Societies, New York (June 2002)

3. Ye, W., Heidemann, J., Estrin, D.: Medium Access Control with Coordinated Adaptive Sleeping for Wireless Sensor Networks. Transactions on Networking, IEEE/ACM 12(3) (June 2004)
4. Dam, T.V., Langendoen, K.: An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks. In: SenSys. Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, Los Angel, CA (November 2003)
5. Jamieson, K., Balakrishnan, H., Tay, Y.C.: Sift: A MAC Protocol for Event-Driven Wireless Sensor Networks. MIT-LCS-TR-894 (May 2003)
6. Arisha, K.A., Youssef, M.A., Younis, M.F.: Energy-Aware TDMA-Based MAC for Sensor Networks. In: IMPACCT. Proceedings IEEE Workshop on Integrated Management of Power Aware Communications, Computing and Networking, New York (May 2002)
7. Lu, G., Krishnamachari, B., Raghavendra, C.S.: An Adaptive Energy-Efficient and Low-Latency MAC for DATA Gathering in Wireless Sensor Networks. In: IPDPS 2004. Proceedings of the 18th International Parallel and Distributed Processing Symposium, Santa Fe, New Mexico (April 2004)
8. Deb, B., Bhatnagar, S., Nath, B.: A Topology Discovery Algorithm for Sensor Networks with Applications to Network Management. In: DCS Technical Report DCS-TR-441, Rutgers University (May 2001)
9. Ai, J., Kong, J.F., Turgut, D.: An Adaptive Coordinated Medium Access Control for Wireless Sensor Networks. In: ISCC. Proceedings of the 9th International Symposium of Computers and Communications (July 2004)
10. Xia, Z.-J., Chen, C.-J.: An Adaptive MAC Protocol for Wireless Sensor Networks. Journal of Beijing Jiaotong University 30(2) (April 2006)
11. Shi, H.-J., Zhang, H.-B., Qian, L., Song, W.-T.: Latency control in S-MAC for WSN. Transactions of Information Technology (2) (2006)

A Scalable Power-Efficient Data Gathering Protocol with Delay Guaranty for Wireless Sensor Networks

Zuzhi Fan[1,2] and Huaibei Zhou[1,3]

[1] Advanced Research Center of Science and Technology, Wuhan University
[2] Computer School, Wuhan University
[3] International School of Software, Wuhan University
zuzhiFan@gmail.com, bzhou@whu.edu.cn

Abstract. Power-efficiency and transmission delay are critical for periodical data gathering applications in wireless sensor networks. This paper presents a scalable power-efficient data gathering protocol with delay guaranty (PDGPDG) for wireless sensor networks. The protocol attempts to balance the energy consumption and transmission delay by dividing the entire network into clusters and then organizing clusters as sub-chains. The parallel transmission among different clusters minimizes the delay and routing data along near optimal sub-chain in the clusters reduces the total energy dissipation. PDGPDG is efficient in the ways that it prolongs the lifetime of network, as well as it takes much lower time to finish a transmission round. Simulation results show that it demonstrates about 200% better performance than that of LEACH in terms of network lifetime and improves the *energy*×*delay* metric by a factor of 2~6 compared to PEGASIS.

1 Introduction

Increasing interests in wireless sensor networks and recent advancement of technology have promoted the development of networked sensors. Many research challenges[1] and applications are associated with the wireless sensor networks. In this paper, we focus on periodical data gathering applications, as defined in the literature [2]. In a typical application as such, all sensor nodes transmit the sensed data to a remote base station periodically[2, 3]. Limited by the size and cost constraints, there are various research challenges to be addressed. In particular, energy-efficiency should be considered with high priority [4]. On the other hand, transmission delay[1] is another important factor which should be addressed.

The logical topology, i.e. the structure used for data transportation, of the network may affect many aspects, such as power-efficiency, delay, scalability and fault-tolerance. Also, the complexity of data routing and processing depends on such a logical topology[5,6]. LEACH protocol [2] is an elegant solution where a hierarchical topology is constructed in the network. Data gathering and fusing are integrated into the cluster topology to improve the power efficiency and to prolong the network

[1] In this paper, transmission delay is described as the interval when the first data is transmitted by sensors and when BS has received data collecting from all sensors in a cycle round.

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 221–232, 2007.

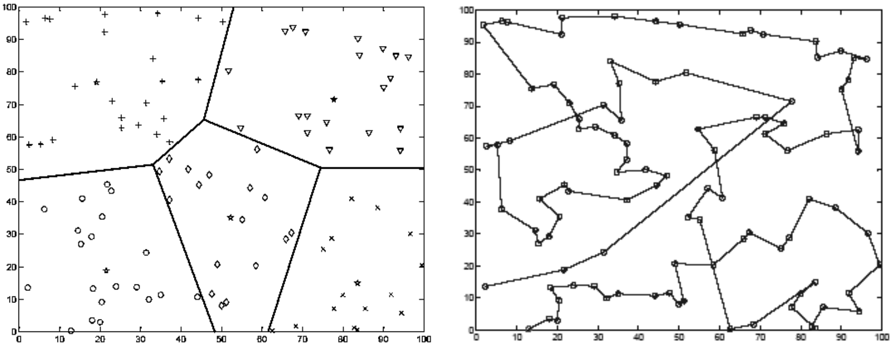
lifetime. In contrast, a near optimal chain-based topology is formed by PEGASIS protocol [3]. In this case, each node only communicates with two other nodes within its proximity to maintain a linear topology. Thus, the power consumption is optimized by reducing the amount of transmission distance. However, the delay in data transportation is greatly increased.

Here, we present a hybrid topology to balance the requirement of power-efficiency and delay constraint in such data gathering scenario. In the topology, the network is divided into multiple clusters and nodes in each cluster are organized as a chain. Based on such a hybrid topology, a scalable data gathering protocol with optimized *energy*×*delay* in comparison with LEACH and PEGASIS is presented.

The reminder of the paper is organized as follows. In section 2, we introduce the related work. In section 3, we firstly analyze the energy conservation in wireless sensor networks, and then present our protocol. At the end of the part, we theoretically analyze and compare the performance among LEACH, PEGASIS and PDGPDG in respect of *energy*×*delay* metric. Section 4 discusses the simulation results. Finally, we conclude the paper and provide some ideas for future in section 5.

2 Related Work

Many data communication protocols have been proposed to achieve energy efficiency, such as LEACH[2], PEGASIS[3], HEED[7], SPIN[8]. LEACH [2] organizes the network into a hierarchy as in Fig.1(a) as follows. First, each node take turns to send data during a designated timeslot to their nearest cluster-head and then keeps asleep for the rest of the cycle time. Second, a cluster-head fuses the collected data before eventually transferring them to the base station. Third, only a small number of cluster-heads send data to the remote base station directly to avoid excessive of long-haul transmissions. Last, LEACH uses random rotation of cluster-heads to avoid the early exhaustion of some nodes. Therefore, LEACH has been proved as an energy-efficient scheme compared to direct transmission.



(a) (b)

Fig. 1. (a) Hierarchical topology for sensor networks; (b) Chain topology for sensor networks

Stephanie Lindsey et al. [3] observe that power consumption can be optimized further by forming the sensors as a near-optimal chain topology, which is called PEGASIS (Fig.1(b)). In PEGASIS protocol [3], each node communicates only with the neighboring nodes on the chain at any time. At last, a designated node will transmit the data to base station by data fusion. Therefore, power consumption can be optimized by reducing the total transmitting distance and the number of long-haul transmissions. Simulation results show that PEGASIS performs better than LEACH by 100 to 300 percent for different network sizes and topologies [3]. However, there is an increasing delay coming with the improved energy saving[9].

Some applications exert limits on acceptable delay specified as a QoS (Quality of Service) requirement. The delay for a packet transmission is dependent on the channel contention time since the processing and propagation delays are negligible. To reduce delay in this case, one should perform parallel transmissions [9]. A typical is that all sensor nodes send data to BS directly at the same time. However, such communications has two problems. (1) CDMA-based technology should be used to achieve multiple simultaneous wireless transmissions without interference; alternatively, spatial-separated nodes perform simultaneous transmissions. Neither is scalable and feasible for sensor networks with large number of sensor nodes which may be deployed very close to each other. (2) The power consumption can be increased with nodes sending data to the remote base station directly.

To achieve the balance of energy consumption and transmission delay, many routing protocols based on chain topology have been addressed. Stephanie Lindsey et al. propose two schemes using CDMA and non-CDMA sensor nodes [9] to balance the energy and delay cost for data gathering. However, both schemes are theoretical and non-scalable. In the literature [10], the authors addressed the energy-efficient data broadcasting and gathering in wireless sensor network by organizing a multiple-chain topology. Furthermore, an energy-efficient chain formation algorithm is proposed, which can be the supplement for our paper. Tri Pham et al [11] presented an enhanced LEACH and a cluster-PEGASIS for energy-efficiency. Moreover, a new metric, data gathering quality (DAQ) is derived for evaluating data aggregating process. Nahdia Tabassum et al [12] proposed a chain-oriented network topology (COSEN), which is very similar as our work. However, their cluster-heads are organized as a super chain. In this paper, we use a different cluster formation heuristic to balance transmission delay and evaluate the performance with *energy*×*delay* metric [9].

3 PDGPDG Architecture

The idea in this work is to combine the simultaneous transmissions among clusters in LEACH [2] with the energy-efficient chaining in PEGASIS [3]. Such a simple and efficient scheme can balance the power consumption and delay effectively. In the paper, we have the following assumptions:

- The sensor nodes are homogeneous and constrained with uniform energy; also, there is no mobility.
- Though the BS is fixed and far away from sensor fields, each node can communicate with BS directly by power control.
- The location of sensor nodes is presumed to be known.

3.1 Power Conservation in WSNs

Power consumption of sensor nodes can be ascribed to the following components of a sensor node: processing unit, radio unit, and sensing unit. The activities of processing units include computation and data fusion. The radio unit transmits and receives messages as radio signals. The sensing and/or actuating unit is used to sense or control the environment. Typically, the radio unit consumes much more energy than the other two. The radio unit may have four states: transmitting state (TX), receiving state (RX), idling state (IDLE) and sleeping state (SLEEP). According to the talk of Estrin D. at the Mobicom2002 tutorial on wireless sensor networks[13], the power consumption in these states are different, which is described as $E_{Tx} \approx E_{Rx} \approx E_{IDLE} \gg E_{SLEEP} \approx E_{Sensor} \approx E_{CPU}$. Here, $E_{Tx}, E_{Rx}, E_{IDLE}, E_{SLEEP}, E_{Sensor}, E_{CPU}$ denote the energy consumption of sensor nodes in the state of transmitting, receiving, idle, sleep, sensing and processing, respectively.

In a scenario of periodical data gathering application[2], the power consumption of sensor nodes during a period T, denoted by E_T, can be broken up into several parts as equation (1).

$$E_T = \sum E_{Tx} + \sum E_{Rx} + E_{IDLE} + E_{SLEEP} + E_{Sensor} + E_{CPU} \quad . \tag{1}$$

The equations (2) used to compute the transmission energy costs and receiving energy costs for a *k*-bit message by a distance of *d* as described in the literature [2].

$$\begin{aligned} E_{Tx}(k,d) &= E_{Tx-elec}(k) + E_{Tx-amp}(k,d) \\ E_{Tx}(k,d) &= E_{elec} * k + \varepsilon_{amp} * k * d^{n} \quad (2 \le n \le 4) \end{aligned} \quad . \tag{2}$$

Here, *n* is the power attenuation exponent which is related to the channel propagation model. For receiving, we have

$$\begin{aligned} E_{Rx}(k) &= E_{Rx-elec}(k) \\ E_{Rx}(k) &= E_{elec} * k \end{aligned} \quad . \tag{3}$$

With $E_{Tx} \approx E_{Rx} \approx E_{IDLE} \gg E_{SLEEP} \approx E_{Sensor} \approx E_{CPU}$ in consideration, we deduce from Eq. (1)(2)(3) that energy consumption can be reduced in several ways. (1) Keep as many as nodes as possible in the sleeping state to avoid unnecessary listening; (2) Reduce the amount of data transmission data by data fusion or aggregation; (3) Avoid long-haul transmission. LEACH[2] and PEGASIS[3] are the representative solutions.

3.2 Our Model

First, we describe a new hybrid topology for sensor networks as in Fig.2, where the sensor field is divided into clusters and each cluster is organized as a sub-chain. Then, based on such a logical topology, a scalable data gathering protocols, named PDGPDG is proposed.

In PDGPDG, the network lifetime is divided into rounds as in previous work [2, 3]. Each round contains topology-setup phase and steady-status phase. In the topology-setup phase, the whole network is organized as a hybrid topology as follows (Fig.2).

In the steady-status phase, each sensor node transmits data periodically in the designated slot along the sub-chain. At last, cluster-heads will aggregate the collected data and transmit it to remote BS.

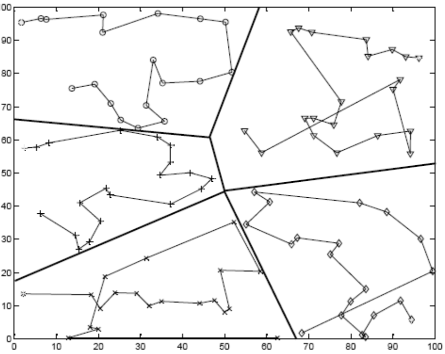


Fig. 2. A hybrid topology with sub-chain in each cluster

In this paper, a central control algorithm is used to form the clusters and sub-chains in the topology-setup phase. In the boot-strap, each sensor node informs the BS of it current energy level and own location, which are assumed to be known. After receiving all information from sensor nodes, the base station (BS) divides the whole network into clusters and selects corresponding cluster-heads. To achieve the balance of energy consumption and transmission delay, the selection of cluster-heads is significant. Here, we chose cluster-heads to meet the following two criteria.

1. The number of sensor nodes in each cluster is approximate to be even. According to prior discussion, the transmission delay depends on the channel contention time. In the cluster, each node sends data in the designated time along the sub-chain, which is contention-free. Therefore, the transmission delay of each cluster is directly related to the number of nodes in the cluster. Moreover, the transmission delay of whole network depends on the longest sub-chain in the field. To minimize the transmission delay, all clusters should keep the same number of nodes as possible.
2. The total sum of distance of sensor nodes along the chains should be minimized. As discussed in section 3.1, the energy dissipation of transmitter is directly related to the distance between the transmitter and receiver. Therefore, each sensor node should transmit data to the nearest neighbor node to save energy.

However, finding *k* optimal clusters is NP-hard problem [14], which is modeled as a global optimization problem in this paper. Here, BS finds cluster-heads using the simulated annealing algorithm [15]. We assume that *N* sensor nodes are divided into *k* clusters and the number of sensor nodes is C_1, C_2, …, C_k. The above optimization criteria can be modeled as following respectively.

$$\min \sum_{i=1}^{k} (C_i - N/k) \qquad C_i \in \{1,...,N\},\ i=1,...,k. \tag{4}$$

$$\min \sum_{i=1}^{k} \Big(\sum_{j=1}^{C_i-1} Dist_{S_j,S_{j+1}} + Dist_{CH_i,BS} \Big) \qquad C_i \in \{1,...,N\},\ i=1,...,k. \tag{5}$$

Here $\mathrm{Dist}_{\mathrm{Sj,Sj+1}}$ denotes the distance between the sensor nodes (S_j) and its neighbor nodes (S_{j+1}) on the chain and $\mathrm{Dist}_{\mathrm{CHi,BS}}$ denotes the distance from the cluster head (CH_i) to the BS. To keep the delay on a higher priority, we assume that the condition (4) is imperative and the condition (5) is supplement.

Once the cluster-heads are selected, each non cluster-head node will select the nearest cluster-head to join. In each cluster, the farthest node from cluster-head will be selected as the start node of the sub-chain. Then, the start node will select the nearest neighbor node of the cluster as its downstream nodes. Iteratively, the sub-chain is constructed using a greedy algorithm as PEGASIS [3]. As the chain structure is formed, the time schedule of each node is decided. In the steady-status phase, each sensor node receives from upstream node and sends the fused data to downstream node in the scheduled timeslot along the chain. To avoid inter-cluster interference, each node on the different sub-chain in PDGPDG communicates using different direct-sequence spread spectrum (DSSS) as in LEACH [2]. At last, the cluster-heads transmit the aggregated data to BS in CDMA way. To balance the energy consumption of cluster-heads, the BS will select new cluster-heads in a new round.

In this paper, we assume infinite data aggregation mode allowed in previous studies [2,3]. Here, the nodes on each sub-chain encircle the cluster head and keep location-related, which is benefit to data aggregation. According to the discussion [11], the chain form algorithm in this paper is better than that of PEGASIS and COSEN [12] in respect of data aggregation quality(DAQ).

The above centralized topology formation algorithm has two benefits. On one head, the BS can create better clusters than distributed one. On the other head, the overhead of clustering process is burden by the BS which is assumed to be powerful and rechargeable. However, distributed topology formation can also be used in this protocol but the overhead of cluster formation is increasing. Therefore, PDGPDG with distributed topology formation algorithm is suitable for the heterogeneous sensor network in which there is a number of more powerful sensor nodes [16].

In this paper, we assume that the sensor nodes are immobile. However, there is topology dynamic in the process of transmission because of the power exhausted or device failure. In the case, the sub-chain will be broken. For the synchronization, we assume a simple way to the issue. If one sensor node in the broken sub-chain can not receive data from upstream node timely, it will fall asleep and wake up until the next round to avoid energy wastage.

From the energy consumption perspective, this scheme performs as well as PEGASIS at least and better than LEACH by forming sub-chains within clusters. From the delay perspective, this scheme is as good as LEACH and better than PEGASIS by parallel transmission.

3.3 *Energy*×*Delay* Analysis for Date Gathering Protocols

In the section, we theoretically analyze the *energy*×*delay* metric per cycle round in data gathering process. The *energy*×*delay* is defined as the product of dissipated energy and transmission delay [9], which is a comprehensive and effective way to evaluate the performance of data gathering protocols.

The *energy*×*delay* cost in a random-deployed sensor network with *N* nodes will depend on the node distribution and density. Consider an example network where the

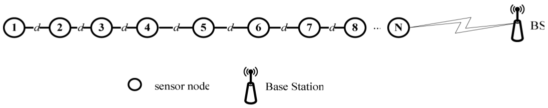


Fig. 3. The example line topology of sensor networks

N nodes are distributed in a straight line with equal distance of *d* between each pair of nodes. BS is located far away from network as in Fig. 3

Because the BS is far distance from all nodes, the energy cost of direct transmission to the BS is assumed the same for all cluster-heads. In LEACH, the total sum of square distance from sensor nodes to their corresponding cluster-head can be deduced as following equation (6).

$$D = \left[(l - \frac{N+1}{2})^2 + \frac{(N^2-1)}{4} \right] \times d^2 \cdot \qquad (6)$$

Here, l $(1 \le l \le N)$ is the ID of cluster head in the line topology. Obviously, the function gets the minimum value while l is equal to $\left\lfloor \frac{N+1}{2} \right\rfloor or \left\lceil \frac{N+1}{2} \right\rceil$ and the maximum value while l is equal 1 or N. For simplicity, we assume that each cluster contains the same number of nodes. For example, the total sensor field is divided into k ($k \ge 1$) clusters and each cluster has N/k sensors. According to equation (6), the maximized energy consumption for each cluster will be the situation in which each cluster-head is located in the end of the line topology. Besides, there will be a single transmission to the BS for each cluster, whose energy consumption depends on the distance from cluster-head to BS. We simplify this additional cost as d and get the maximized energy consumption for each cluster in LEACH as $\left(\left\lfloor \frac{N^2}{2k^2} \right\rfloor + \frac{N}{k} \right) \times d^2$ .With k clusters and the delay of N/k units into consideration, the maximized *energy*×*delay* for the LEACH is $\left(\left\lfloor \frac{N}{2k^2} \right\rfloor + \frac{1}{k} \right) \times d^2 \times N^2$. In the same way, the minimized energy consumption will be the situation in which each cluster-head is located in the middle of line topology. The minimized *energy*×*delay* can be evaluated as $\left(\left\lfloor \frac{N}{4k^2} \right\rfloor + \frac{1}{k} \right) \times d^2 \times N^2$. At last, the *energy*×*delay* for LEACH meet the requirement as following equation (7).

$$\left(\left\lfloor \frac{N}{4k^2} \right\rfloor + \frac{1}{k} \right) \times d^2 \times N^2 \le Energy \times Delay \le \left(\left\lfloor \frac{N}{2k^2} \right\rfloor + \frac{1}{k} \right) \times d^2 \times N^2 \ . \qquad (7)$$

For comparison, we introduce the coefficient ε , which is defined as follows.

$$\left[\left(\left\lfloor \frac{N}{4k^2} \right\rfloor + \frac{1}{k} \right) \le \varepsilon \le \left(\left\lfloor \frac{N}{2k^2} \right\rfloor + \frac{1}{k} \right) \right] \cdot \qquad (8)$$

Therefore, the *energy*×*delay* cost of LEACH can be denoted as $\varepsilon \times N^2 \times d^2$.

In the same way [9], the energy consumption of each sub-chain in PDGPDG protocol can be calculated as $(\frac{N}{k}-1)\times d^2$. With the energy of transmission to BS and a rough delay of N/k units into consideration, the total *energy*×*delay* cost of PDGPDG can be calculated as $1/k\times d^2\times N^2$. According to the discussion in the literature [9], the *energy*×*delay* cost of PEGASIS will be $N^2\times d^2$. From above analysis, the coefficient of *energy*×*delay* cost is $1, \varepsilon$, $1/k$ for PEGASIS, LEACH and our scheme, respectively.

As per Equation (8) and $k\geq 1$, we can conclude that our scheme has least *energy*×*delay* cost among three schemes in the case of line topology. However, it is difficult to mathematically analyze the cost for stochastic distributed nodes and we will use simulations to evaluate it.

4 Simulation Results and Analysis

In order to analyze the performance of PDGPDG comprehensively and accurately, we extends network simulator ns-2 [17] to support PEGASIS and PDGPDG protocol. Several simulation parameters are selected to test and compare these schemes in terms of network lifetime, the received data at BS per unit energy and *energy*×*delay*. The *energy*×*delay* cost is the primary metric of evaluation. For justice, we implement the centralized version of PEGASIS, PDGPDG, named PEGASIS-C, PDGPDG-C respectively and compare them with LEACH-C, which is also the centralized implementation of LEACH algorithm [18].

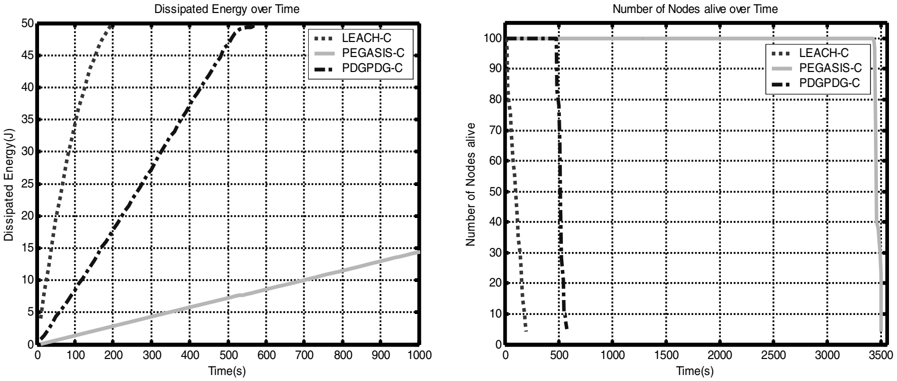
4.1 Performance Comparison among LEACH, PEGASIS and PDGPDG

In the section, we compare the three schemes in terms of energy dissipation, the number of nodes alive, the amount of transmitted data per unit energy and *energy*×*delay*. Most of simulation parameters will be the same as LEACH protocol [2] except that the initial energy of each node is 0.5 Joules here. The dimension of simulation is 100mx100m with BS at (50,175).

Fig.4(a) shows that the lifetime of our scheme increases by about a factor of 2 than LEACH and PEGASIS is better than LEACH by about a factor of 17. In Fig.4(b), the nodes in LEACH protocol die quickly, but the nodes in our scheme and PEGASIS can keep for a long and stable time, which demonstrates that the individual sensor node in chain-based topology has more balanced energy dissipation than those in cluster-based topology.

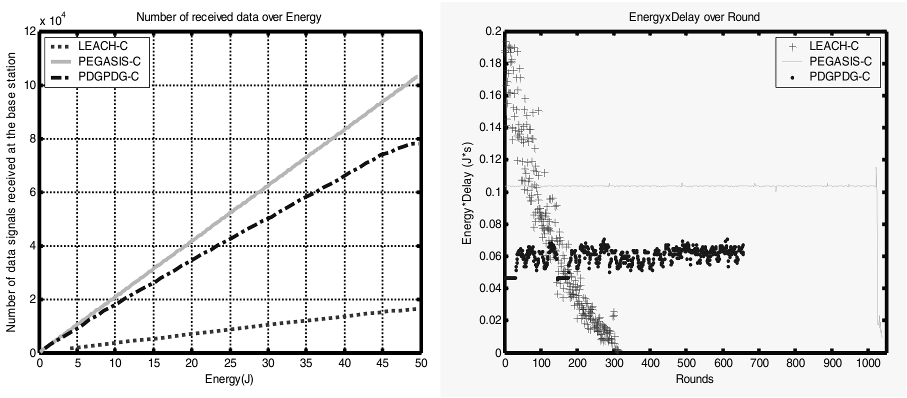
Fig.5(a) describes the efficiency of received data per energy cost. It shows that PEGASIS performs better than PDGPDG. However, PDGPDG is better than LEACH in the simulation, which proves that energy efficiency in the chain-based topology is much higher than cluster-based topology.

Fig.5(b) provides a detailed description of the *energy*×*delay* performance evaluation over rounds for three protocols. Different from the *energy*×*delay* metric in the literature[9], the *energy*×*delay* here is the product of practical energy consumption and transmission delay in each round, but not the statistical data. We can conclude that the *energy*×*delay* in LEACH decreases quickly with the time since sensor nodes



(a) (b)

Fig. 4. (a) The total dissipated energy over time. (b) The number of nodes alive over time.



(a) (b)

Fig. 5. (a) Total amount of data signals received at the BS over energy, where each node begins with 0.5J of energy. (b) *energy*×*delay* performance over round.

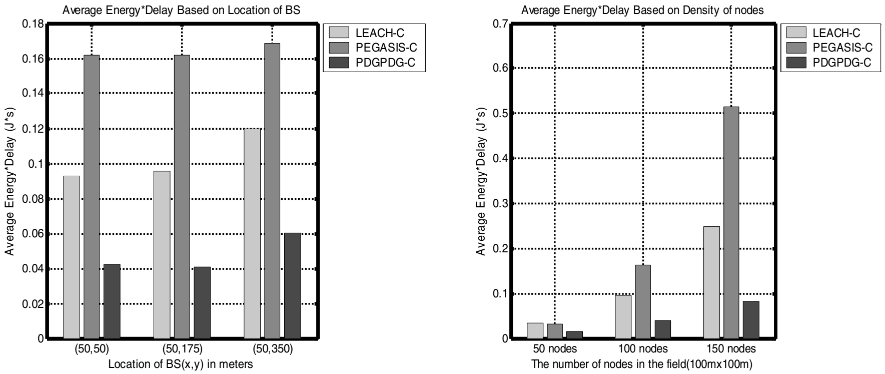
in LEACH dissipate energy more quickly than PDGPDG and PEGASIS. Moreover, most of nodes die quickly. However, the *energy*×*delay* in PEGASIS keeps a high level because of high delay cost. The *energy*×*delay* in PDGPDG is rather low compared to LEACH and PEGASIS, which verifies that our scheme make a good balance between power consumption and transmission delay.

We also observe that, the *energy*×*delay* in PEGASIS is stable but the experimental results for PDGPDG and LEACH fluctuate from round to round. Because the chain topology in PEGASIS keeps unchanged in the whole lifetime, both energy consumption and transmission delay in each round keep constant. However, for LEACH and PDGPDG protocols, BS will select new cluster-heads and reform the network topology at fixed interval of time, which has influence on both energy dissipation and transmission delay in next round.

4.2 Evaluation of Network Parameters

In the section, the influence of two parameters, the location of BS and sensors density, on the performance of three protocols is verified.

Fig. 6(a) shows the results for three protocols based on different BS locations. Our scheme is always more 2 times better than LEACH and 4 times better than PEGASIS for a 100mx100m network in terms of average *energy*×*delay* with 100 nodes deployed when BS is located at far or near position.



(a) (b)

Fig. 6. (a) Average *energy*×*delay* for a 100mx100m network with BS locations at (50, 50), (50, 175), (50, 350) respectively. (b) Average *energy*×*delay* for a 100mx100m network with BS (50, 175) and the number of nodes is 50,100,150 respectively.

Fig. 6(b) shows that, PDGPDG achieves approximate 2, 2, 4 times improvement than LEACH and 2, 4, 6 times improvement than PEGASIS in terms of average *energy*×*delay* with 50,100,150 nodes deployed in a 100mx100m network respectively. These simulation results demonstrate that:

1. As the BS moves farther away from the sensor fields, the energy consumption of long-haul transmission from cluster-heads to BS is increasing. Therefore, the average *energy*×*delay* is increasing for all three schemes. While there is only one node transmitting data to BS in PEGASIS, the metric increases more slowly compared to LEACH and PDGPDG. However, PDGPDG performs much better than LEACH and PEGASIS in all cases.
2. With the node density increasing, the average *energy*×*delay* for all schemes increases too. However, our scheme performs the best in three schemes.
3. Our scheme performs much better than the other two schemes in high-densely deployed sensor networks in terms of *energy*×*delay*. Therefore, PDGPDG is more suitable for high-density network.

5 Conclusion

In the paper, we present a hybrid topology for wireless sensor networks. In the topology, sensor nodes are divided into clusters and then organized as sub-chains.

Based on the topology, a scalable power-efficient data gathering protocol with delay guaranty, PDGPDG is proposed. In both analysis and simulations, PDGPDG outperforms LEACH in energy dissipation and PEGASIS in transmission delay respectively. Also, our simulation results show that PDGPDG performs better than both LEACH and PEGASIS in terms of *energy*×*delay*.

However, to validate the assumptions, more experiments should be done. The distributed version of PDGPDG should be implemented, in which sensor nodes organize themselves as clusters randomly. We also assume that cluster-heads transmit the fused data to BS directly, which will be inefficient in large-scale deployed networks. In such a scenario, cluster-heads which are far away from BS will die quickly for exhausted energy. In our future work, we will consider the model of multi-hop communication among cluster-heads[19].

References

1. Chee-Yee, C., Kumar, S.P.: Sensor networks: evolution, opportunities, and challenges. In: Proceedings of the IEEE, pp. 1247–1256 (2003)
2. Heinzelman, W., Chandrakasan, A., Balakrishnan, H.: Energy-efficient communication protocol for wireless microsensor networks. In: HICSS. Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, Piscataway, NJ, USA (2000)
3. Lindsey, S., Raghavendra, C.S.: PEGASIS: Power-efficient gathering in sensor information systems. In: Proceedings of IEEE Aerospace Conference, pp. 1125–1130 (2002)
4. Akyildiz, I.F, Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless Sensor Networks: A Survey. Computer Networks 38(4), 393–423 (2002)
5. Tilak, S., Abu-Ghazaleh, N.B., Heinzelman, W.: A taxonomy of wireless micro-sensor network models. ACM Mobile Computing and Communications Review 6(2), 28–36 (2002)
6. Fan, Z.Z., Yu, Q.C., Zhou, H.B.: Mixed Type Logical Topology for wireless sensor networks. In: Proceedings of International Conference on Wireless Communications, Networking and Mobile Computing, Wuhan, China (2005)
7. Younis, O., Fahmy, S.: HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks. IEEE Transactions on Mobile Computing 3(4), 366–379 (2004)
8. Rabiner, H.W., Joanna, K., Hari, B.: Adaptive protocols for information dissemination in wireless sensor networks. In: Proceedings of the 5th annual ACM/IEEE International Conference on Mobile Computing and Networking, Washington USA (1999)
9. Stephanie, L., Cauligi, R., Krishna, M.S.: Data Gathering Algorithms in Sensor Networks Using Energy Metrics. IEEE Transactions on Parallel and Distributed Systems 13(9), 924–935 (2002)
10. Kemei, D., Jie, W., Dan, Z.: Chain-based protocols for data broadcasting and gathering in the sensor networks. In: Proceedings of International Parallel and Distributed Processing Symposium (2003)
11. Pham, T., Jik, K.E., Moh, M.: On data aggregation quality and energy efficiency of wireless sensor network protocols - extended summary. In: First International Conference on Broadband Networks (2004)
12. Tabassum, N., Ehsanul, Q., Mamun, K., Urano, Y.: COSEN: A Chain Oriented Sensor Network for Efficient Data Collection. In: ITNG 2006. Third International Conference on Information Technology: New Generations (2006)

13. Estrin, D.: Tutorial Wireless Sensor Networks Part IV: Sensor Network Protocols. In: MobiCom (2002)
14. Pankaj, K.A., Cecilia, M.P.: Exact and approximation algorithms for clustering. In: Proceedings of the ninth annual ACM-SIAM symposium on Discrete algorithms, San Francisco, California, United States (1998)
15. Lai, K.K., Jimmy, W.M.C.: Developing a simulated annealing algorithm for the cutting stock problem, vol. 32, pp. 115–127. Pergamon Press, Inc, New York (1997)
16. Duarte-Melo, E.J., Liu, M.: Analysis of energy consumption and lifetime of heterogeneous wireless sensor networks. In: IEEE Global Telecommunications Conference, New York, USA, pp. 21–25 (2002)
17. UCB/LBNL/VINT Network Simulator–ns (2000), http://nsnam.isi.edu/nsnam
18. Heinzelman, W.: Application-Specific Protocol Architectures for Wireless Networks. PhD thesis. In Dept. of Electrical Eng. and Computer Science, Massachusetts Inst. of Technology (2000)
19. Devendar, M., Fei, D., Xiaojiang, D., Chao, Y.: Load Balance and Energy Efficient Data Gathering in Wireless Sensor Networks. In: MASS. IEEE International Conference on Mobile Adhoc and Sensor Systems, pp. 586–591 (2006)

An Efficient Rate-Adaptive MAC for IEEE 802.11

Yuanzhu Peter Chen[1], Jian Zhang[2], and Anne N. Ngugi[1]

[1] Department of Computer Science, Memorial University of Newfoundland, Canada
[2] CAIP, Rutgers University, USA

Abstract. Data rate selection for IEEE 802.11-based wireless networks is not specified in the Specification. The problem of determining an appropriate data rate for the sender to send DATA frames and to adapt to changing channel conditions is referred to as *rate adaptation*. We propose DRA (differential rate adaptation), a rate adaptation scheme for IEEE 802.11 networks. It enables a high network throughput by adaptively tuning the data transmission rate according to the channel conditions. It is responsive to link quality changes and has little implementation overhead. Our experiments indicate that DRA yields a throughput improvement of about 20% to 25% compared to previous work.

1 Introduction

Mobile communication is becoming an integral component of a new era of life style. Along with other forms of wireless networks, mobile ad hoc and mesh networks provide a flexible yet economical platform for short- and mid-range communications. The most prevalent technology to implement these networks is the IEEE 802.11 compliant devices. The PHY layer of the IEEE 802.11 standard family provides a set of different modulation schemes and data rates for use in different channel conditions. The standard itself, however, does not specify how these data rates are selected adaptively. Therefore, the problem of determining an appropriate modulation scheme and thus a reasonable data rate attracts interests in the research of wireless networking.

Existing rate adaptation schemes in the literature generally fall into two categories. At one extreme, there are "open-loop" solutions that allows the sender to continually probe higher data rate. The ARF of Lucent WaveLAN-II [1] is an example. At the other extreme, there are "closed-loop" solutions where a sender explicitly solicits the receiver to provide reception quality information to determine an appropriate data rate. Examples of this latter approach include RBAR [2] and OAR [3]. To avoid mistaken rate reduction at the sending side and, thus, to further improve the performance of the closed-loop rate-adaptation schemes as above, a sender should differentiate the causes of lost DATA frames to reach more informed decisions, such as in LD-ARF [4] and CARA [5]. The closed-loop approaches intend to solve the *blind probing* of the open-loop approaches, where a sender keeps trying sending a DATA frame at a higher data rate from time

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 233–243, 2007.

to time, even though the receiver can not actually handle a faster transmission. The result of such blindness is the loss of frames transmitted at overly high data rates. The cost to overcome such a blindness problem is the mandatory use of the RTS/CTS control frames to measure the channel condition at the receiving side. This introduces extra overhead since RTS/CTS frames can be disabled optionally in the original IEEE 802.11 DCF for higher network throughput.

In this work, we propose a rate adaptation scheme that combines the advantages of the open-loop and the closed-loop approaches, called Differential Rate Adaptation (DRA). In particular, we use a single RTS/CTS exchange between a given sender-receiver pair to lead multiple DATA/ACK dialogs in the sequel. Each ACK contains in its header a bit to indicate the sender if the next higher data rate is recommended or not according to the reception of the previous DATA frame. Use of this feedback to the sender also provides a precision tolerance of the earlier channel quality estimation via RTS/CTS. Such a design follows a similar rationale of the Explicit Congestion Notification (ECN) as the TCP/IP architecture. The benefit of doing so is to avoid undesired outcomes before they happen rather than recovering from bad situations after they have occurred. In case of a lost DATA frame, the retransmit may be done at a lower rate.

2 Related Work

In wireless communications, rate adaptation is a mechanism for the sender to determine an appropriate data transfer rate to use the channel to the maximum extent. Due to the transient nature of channel conditions, such a mechanism must be responsive to the changes with a small overhead. Here, we focus on rate adaptation mechanisms proposed for the IEEE 802.11 based wireless networks, infrastructured or ad hoc. Since rate adaptation is not part of the IEEE 802.11 Specifications [6], the design of these mechanisms varies considerably. Depending on the scope of information that a sender uses to make the decision on rate selection, these mechanisms are usually divided into two categories, open loop and closed loop. In an open loop approach, the sender makes the decision solely based on its own perception, such as the outcome of a previous DATA transmission or the reception quality of an ACK. In a closed loop design, the sender explicitly solicits the receiver to estimate the channel condition and to feed this information back to the sender to select an appropriate data rate. In this section, we review some typical proposals of rate adaptation, open loop followed by closed loop.

The first and most widely adopted open loop rate adaptation in 802.11 devices is the ARF (auto-rate fallback) of WaveLAN-II of Lucent Technologies [1]. It consists rate probing and fallback, an idea similar to various TCP congestion control protocols. After a certain number (10 by default) of consecutive successful DATA transmissions at a given data rate, and if there is a higher data rate available, the sender selects a higher rate for the subsequent transmissions. If the channel can sustain the higher rate for a number of DATA frames, the next higher data rate is probed. If, however, a DATA transmission fails (one retrial is allowed for each data rate by default), the sender falls back to a lower data rate to retransmit the same DATA frame. In this case, a further fallback will be needed if the retrial of the transmission fails, too. ARF is simple and works fairly well. Some variants of ARF have been proposed, e.g. the FER (frame error rate) based approach [7]. In addition to using purely link level observations such as the outcomes of DATA frames, some other open loop proposals go further to use information provided by the PHY layer, such as SINR (signal to interference and noise ratio) or RSS (received signal strength) [8,9]. An important but reasonable assumption of these protocols is the symmetry of channel conditions. As a result, the sender can look up a good data rate from a pre-established table based on the SINR or RSS provided by its own PHY layer, hoping that the receiver is experiencing something similar.

Thus far, the open loop mechanisms implicitly assume that the loss of a DATA frame is caused by bad channel conditions and can be relieved by reducing the data rate. However, in a dense, especially multi-hop, wireless network, this can be well caused by collisions. Reducing data rate regardless of its actual causes not only brings down the network throughput but can also cause further collisions due to a longer transmission time of the same DATA frame. Observing this, LD-ARF (loss-differentiating ARF) [4] and CARA (collision-ware rate adaptation) [5] are proposed as smart rate fallback mechanisms by differentiating the causes of a lost DATA frame. Both LD-ARF and CARA probe for higher data rate as earlier open loop proposals. But when losing a DATA frame, the sender falls back to a lower rate only if it believes that the DATA loss was caused by a bad channel; otherwise, it simply retransmits the frame at the same data rate. LD-ARF and CARA differ in the way that they deduce the causes of a lost DATA frame. In LD-ARF, two loss differentiation methods are used, depending on whether RTS/CTS is used. In the RTS/CTS mode, the loss of a DATA frame after a successful reception of a CTS frame is considered to be caused by bad channel conditions. This is because of 802.11's robustness in the transmitting control frames, both in terms of modulation and duration. In this case, a lower data rate should be used. On the other hand, if an expected CTS is missing, the DATA frame should be transmitted at the same data rate because of the collision signified by the lost RTS. In the basic mode where RTS/CTS is disabled, and also assuming that there are no hidden terminals, a garbled DATA frame trigger the receiver to transmit a NAK (negative acknowledgment) if the MAC header of the frame can be reconstructed correctly. The rationale for LD-ARF is that, even when the channel condition is so bad that the entire frame is garbled, the MAC header can still be intact because of its small length, given a fixed BER. Thus, a NAK signifies the sender of bad channel conditions while losing a DATA frame without a NAK coming back indicates a collision. CARA also has two methods to detect collisions. The first one is similar to that of LD-ARF. That is, a successful RTS/CTS exchange followed by a lost DATA frame indicates a bad channel condition. Realizing the communications overhead of enabling RTS/CTS, CARA employs an RTS activation mechanism. The RTS/CTS are used only occasionally for diagnostic purposes.

The overhead of the open loop design is small in terms of extra bandwidth consumed. Another advantage is that it does not require the modification of the frame forms defined by the Specifications. On the flip side, the information scope used by any open loop approach is limited. A sender draws a decision upon its own perception, may it be from the link layer or physical layer. In addition, the constant attempt to transmit at a higher data rate can affect the network throughput negatively. Loss differentiation may improve the performance to a degree by avoiding reducing data rate mistakenly, but the deduction of frame collision is not sufficiently accurate, especially when there are hidden or masked nodes. Closed loop approaches attempt to make more informed decisions with the help of the receiver. Indeed, whether a DATA frame can be received correctly at a given data rate can only be estimated much more precisely on the receiving side. The cost of transferring the information from the receiving side is an increased overhead in protocol implementation.

RBAR (receiver-based auto rate) [2] is the first closed loop protocol in the context of IEEE 802.11 networks. In RBAR, a sender always transmits an RTS frame before transmitting a DATA frame. Upon receiving the RTS, the receiver also measures the SINR of the moment. Based on acceptable BER, the receiver looks up the highest data rate that the transient SINR supports. This data rate is fed back to the sender using a modified CTS frame. OAR (opportunistic auto-rate) [3] enhances RBAR using fragmentation of the IEEE 802.11 MAC. It improves the efficiency of RBAR significantly by allowing a single RTS/CTS exchange to lead a train of DATA/ACK pairs. This overcomes the major disadvantage of low efficiency of the closed loop design. This idea is further extended by MAD (medium access diversity) [10]. MAD is designed to improve network throughput by allowing a sender to choose a neighbor that can receive a DATA frame at the highest data rate. In essence, MAD uses a link level anycast mechanism, where the RTS format is extended to include a list of multiple receiver addresses. Such an extension is also made to solve the HOL (head of line) blocking problem in mesh networks, referred to as MRTS (multicast RTS) [11].

Using additional information from the receiving side, the closed loop design usually is more responsive to channel condition changes. Its overhead can be reduced by allowing a data burst after a single RTS/CTS dialog. Still, it is susceptible to inaccurate channel estimation and the length of the data burst is heavily constrained by the channel coherence time. DRA (differential rate adaptation) proposed in this work combines the advantages of both open and closed loop approaches to achieve better performance, as described in the next section.

3 Design of DRA

DRA uses a single RTS/CTS dialog to lead a burst of DATA/ACK pairs. This is essentially a combination of the closed and open loop designs. The probing of a higher data rate is done with an effective differential compensation, i.e. using a flag in the ACK header, without the risk of using too high a data rate for the channel to sustain. This is done without extra overhead and with maximum

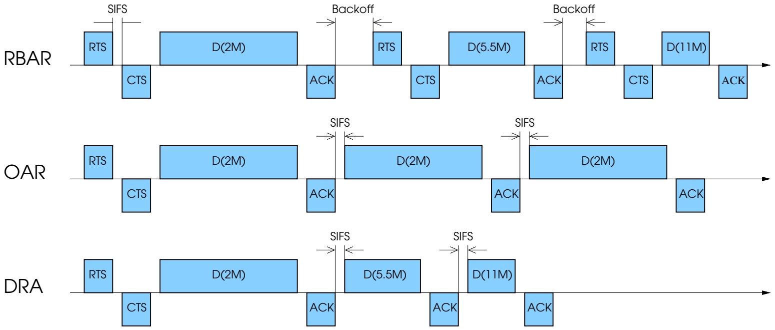


Fig. 1. Design of DRA

compatibility with the Specifications. As a result, during the entire data burst, our single-bit feedback mechanism compensates the moderate channel condition changes, while more significant changes will be captured by the RTS/CTS leading each data burst. Another advantage of DRA is its tolerance in the inaccuracy in channel conditions estimated by the RTS/CTS exchange; the data rate adopted by the sender matches the channel quality better and better as the burst goes on. A schematic comparison between DRA with RBAR and OAR is illustrated in Fig. 1.

3.1 Data Rate Estimation and Feedback — Receiving Side Story

DRA enlists the receiver to feed the channel condition information to the sender to close its control loop. To do that, when the receiver receives the RTS or DATA frame, it also records the SINR when the frame was received. Based on that, the receiver can look up from a table the highest data rate that the recorded SINR supports with an acceptable bit error rate. Then the receiver puts such a planned data rate in the CTS frame so that the sender can adopt this rate in the subsequent burst of DATA frames. Further, the estimation errors and the channel condition changes can be compensated by piggy-backing a single bit in the ACK from the receiver to indicate if the next higher data is feasible for the next DATA frame in the burst.

Once the receiver has determined the data rate r_i, it needs to feed this information back to the sender. To do that, we change the definition of the "duration" field of a MAC header, as in RBAR and OAR. In the Specification, the duration field is a standard 16-bit field of a data or control frame. The value is the amount of time needed before the subsequent ACK is received in milliseconds. This is used to set the NAV (network allocation vector) of a node that overhears the frame to accomplish VCS (virtual carrier sensing). Here, it is changed to two subfields, *rate* and *length*, of 4 and 12 bits, respectively (Fig. 2). The rate subfield is sufficient to address 16 different data rates, which is sufficient to represent the data rates required by 802.11 Legacy, 11b, and 11a/g. (That is, 1M and 2M from Legacy, 5.5M and 11M from 11b, and 6M, 9M, 12M, 18M, 24M, 36M, 48M, and

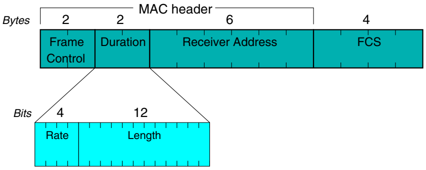


Fig. 2. CTS frame format of DRA

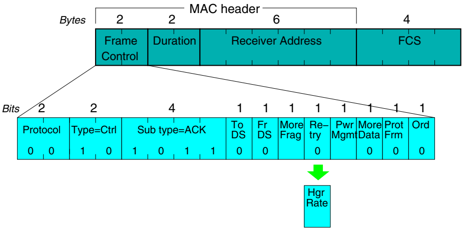


Fig. 3. ACK frame format of DRA

54M from 11a/g) The length subfield contains the length of the MSDU (frame body) in bytes. The 12 bits therein can potentially represent an MSDU of 4096 bytes, which is greater than the maximum body size of 2346 bytes. Using these two subfield, a node that overhears the header can reconstruct the NAV value for VCS. When constructing the CTS frame, the receiver puts the data rate index i in the rate subfield and copies the value of the length subfield in the incoming RTS frame header. It then transmits the CTS frame to the sender at the basic data rate.

Similarly, the receiver also indicates to the sender if a higher data rate should be adopted using a single bit. To do that, we utilize the "retry" bit in the *frame control* field of the MAC header since it is redundant for the ACK frame (Fig. 3). We call such a bit the "higher rate" flag. When the receiver receives a DATA frame, it also estimates the highest data rate r_i that could have been used by that frame transmission. If the channel condition has improved significantly so that r_i is higher than the data rate at which the DATA was received, the higher rate flag is set to 1 to inform the sender that the next higher data rate can be used for the subsequent DATA frame. Otherwise, the flag is set to 0.

3.2 Adaptive DATA Burst—Sending Side Story

In DRA, a sender contends for the channel before exchanging RTS/CTS with the receiver. Then, a burst of DATA/ACK pairs will be transmitted between the sending and receiving parties. The inter-frame space between each of these consecutive frames is SIFS (as in Fig. 1), so that the train of frames will not be interrupted by other nodes. This burst of DATA/ACK frames is responsible for adapting to channel condition changes and for retransmitting garbled packets.

The data rate $R = r_i$ $(1 \leq i \leq k)$ of the first DATA frame is determined by the value set in the "rate" field of the received CTS frame. The sender then constructs the DATA frame and transmits the frame at R Mbps. It then waits for the ACK to indicate if the transmission was successful. If so, it will transmit the next DATA frame; otherwise, it must retransmit the same DATA frame. In the first case, whether a higher data rate should be used for the next DATA frame is determined by the "higher rate" flag in the ACK header. If the flag is set to 1, it sets R to $r_{\min\{k,i+1\}}$, i.e. to the next available higher data rate, to explore for higher throughput. If the flag is 0, it remains at the same data rate

r_i. In the second case, where the expected ACK was missing, the same DATA frame is retransmitted at the same data rate.

In the design of DRA, the temporal length of the burst can be considerably longer than that needed to complete the RTS/CTS/DATA/ACK 4-way handshake at the basic rate due to its adaptiveness to channel conditions. We denote this burst length by T_b. That is, as long as the queue within the sender is non-empty, it keeps transmitting the next DATA frame after SIFS of receiving the ACK. The sender keeps pumping data through the wireless channel until T_b seconds has elapsed since the moment it started sending the first DATA frame of the burst. The choice of T_b should satisfy that, during this amount of time, the channel condition at the receiver can be compensated by the rate adaptation mechanism as described above. In our implementation, T_b is set to 50 ms, which is approximately the time to transmit slightly over two DATA frames of maximum size (2346 bytes) at the basic data rate of 1 Mbps. Such a choice of T_b is verified by the calculation in OAR. In contrast, T_b is set to 20ms in OAR to accommodate one where the coherence time is about 122.88 (24.57, 12.28, 6.14, resp.) ms for a center frequency of 2.4 GHz at mobile speed of 1 (5, 10, 20, resp.) m/s. After completing the burst, the sender must contend for the channel, as specified in the DCF (distributed coordination function) of the Specifications [6], if it has more data to transmit.

3.3 Setting the NAV — Everybody Else

An 802.11 device can be used to implement a multi-hop wireless network. To cope with the hidden terminal problem, the duration information is embedded in each type of frame, so that, whenever a node overhears the frame, it stays away from the channel for the indicated amount of time. That is, the network allocation vector is set to the duration value. As discussed earlier, DRA differs from the Specifications in that its duration field has two components (Fig. 2), the rate index i and the payload size S. Note that the PHY layer header and MAC headers of a frame are all transmitted at the fixed basic rate. Thus, the only variables that contribute the the transmission time of a data frame are the data rate r_i and the size of payload S. We further denote the time needed to transmit the PHY (RTS, CTS, DATA, and ACK, resp.) header by H_p (H_r, H_c, H_d, and H_a, resp.). When a node overhears a frame, it sets if NAV as follows:

- RTS — $SIFS+H_p+H_c$. That is, when overhearing an RTS, the NAV should be set to secure the channel until point A in Fig. 4.
- CTS — $SIFS + H_p + H_d + 8 \times S/r_i$. That is, when overhearing a CTS, the NAV should be set to secure the channel until point B.
- DATA — $SIFS + H_p + H_a$. That is, when overhearing a DATA, the NAV should be set to secure the channel until point C.
- ACK — $SIFS + H_p + H_d + 8 \times S/r_i$. That is, when overhearing an ACK, the NAV should be set to secure the channel until point D.

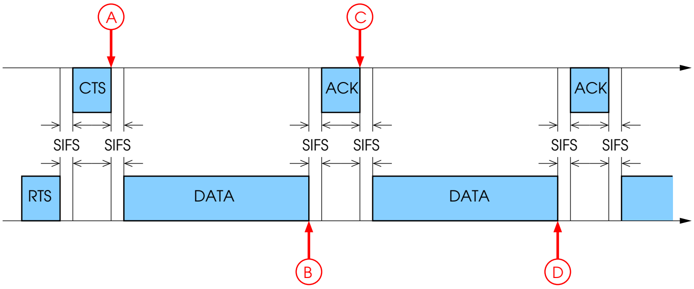


Fig. 4. Setting the NAV

4 Simulation

To study the effectiveness of DRA, we resort to packet level simulation using ns-2. The focus is to study the data link layer throughput of DRA in highly dynamic channel conditions with and without hidden terminals.

We use the pre-computed time series data of Punnroose et al. [12] to simulate a rapidly fading channel that follows the Ricean distribution. In the simulation, we vary K, the Ricean parameter, between 0 and 5 to achieve different levels of contribution of the line-of-sight component in the received signal. Since DRA is an extension of OAR, we compare these two protocols' performance in the same changing channel condition. Our preliminary experiments showed that with relatively low node mobility, say speed of 2.5m/s (setting the maximum Doppler frequency f_m to 30Hz), DRA offers a slightly higher throughput. This also indicates that OAR's succinct design is fairly effective for a low to medium mobility rate. In contrast, DRA's per-fragment rate adaptation achieves higher throughput even if the channel conditions change rapidly. This is verified by our experiment below when setting f_m to 300Hz (i.e. 25m/s of maximum mobility velocity). Furthermore, DRA enables long fragment bursts to reduce the control overhead, thus, having higher efficiency.

The testing is done in two scenarios, without and with hidden nodes (Fig. 5). In each scenario, we set the fragment burst length to 6ms and 50ms, respectively, to find out about the effect of using a longer burst length. Apparently, longer bursts reduce the protocol overhead introduced by the RTS/CTS handshake at the beginning of each burst. However, the changing channel condition will render

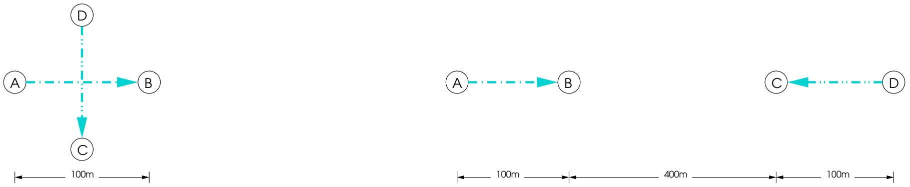


Fig. 5. Simulation scenarios

that the data rate estimated by the receiver and fed back via CTS to be invalid as the burst goes on. Without a compensation mechanism, the burst length can be rather restricted in a rapidly dynamic environment. In our simulation, we observe how DRA benefits from its adaptiveness to support longer bursts well.

In the first scenario (Fig. 5, left), we deploy two CBR flows (A to B and D to C), each of which can saturate the network capacity. The transmitter-receiver separation distance is 100m, such that the data rate fluctuates between 1 and 11Mbps. Since all nodes in this scenario are not farther than 100m apart, they can, in most cases, decode the control frames (i.e. RTS, CTS, and ACK) and the header of DATA frames. Therefore, these nodes are fully connected to each other. For both OAR and DRA, we set the burst length to 6ms and 50ms. We start the two flows at the beginning of the simulation simultaneously. The simulation has a duration of 50 seconds and is repeated 10 times. We measure the number of packets aggregated for the two flows per unit of time. We observed that the measurement stabilized in a short time. For a fixed Ricean parameter K, we plot the total throughput, i.e. number of packet received in 50 seconds, for each protocol-burstlength combination (DRA vs OAR and 6ms vs 50ms) as depicted in Fig. 6. In the plot, we see that the throughput increases as the line-of-sight component becomes stronger (larger K) for each combination. In addition, for the shorter burst length of 6ms, DRA possesses an average of about 4% of throughput gain over OAR. However, for the longer burst length of 50ms, DRA's adaptiveness introduces an average of about 25% of throughput gain.

In the second scenario (Fig. 5, right), we also deploy two CBR flows (A to B and D to C), But here, we separate these flows fairly far away such that the two receivers (B and C) are 400m apart and the senders (A and D) are 600m apart. As a result of the ns-2 default settings, which is fairly typical in this aspect in reality, the two pairs are hidden from each other but the two receivers are still within the carrier sensing range of both senders. Ideally in this case, the two flows should be transported in parallel. But due to the fact that the NAV cannot be set effectively by a distant transmitter, there will not be a 100% parallelism. Simulation done for this scenario is plotted in Fig. 7, and it

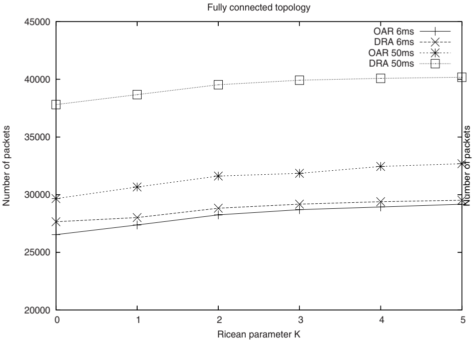


Fig. 6. Full connection

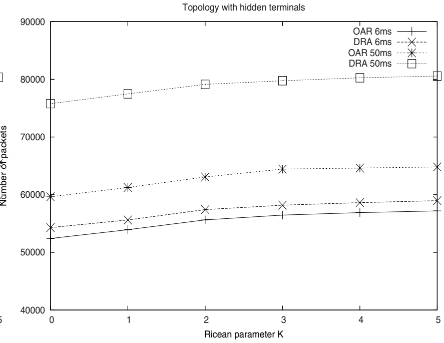


Fig. 7. With hidden nodes

indicates that the effect of hidden transmitters is minimized and the throughput is roughly doubled in the matching point in the previous scenario. Here, the 6ms burst length enables DRA an approximately 3% of throughput gain and the longer 50ms burst length offers a more significant 20% throughput gain.

5 Conclusion and Future Work

In this work, we investigate a data rate selection method for IEEE 802.11 devices. In particular, we present a feedback mechanism, DRA, for a receiver to compensate channel condition changes. This is essentially combining the advantages of the open- and closed-loop designs for the rate adaptation solutions in the literature. The design of DRA has zero extra overhead compared to its ancestor, OAR. In our simulation, DRA indicates a 20% to 25% throughput gain when using a longer burst length. In the research to follow, we plan to integrate loss differentiation to avoid blind rate fallback caused by a lost fragment. Our preliminary work indicated that a simple inclusion of the CCA-based or the transmission history based approaches in the literature does not offer a noticeable improvement of DRA. Nevertheless, more sophisticated decision mechanisms may further improve DRA.

References

1. Kamerman, A., Monteban, L.: WaveLAN-II: a high-performance wireless LAN for the unlicensed band. Bell Labs Technical Journal 2(3), 118–133 (1997)
2. Holland, G., Vaidya, N., Bahl, P.: A rate-adaptive MAC protocol for multi-hop wireless networks. In: Proceedings of MobiCom, Rome, Italy, pp. 236–251 (July 2001)
3. Sadeghi, B., Kanodia, V., Sabharwal, A., Knightly, E.: Opportunistic media access for multirate ad hoc networks. In: Proceedings of MobiCom, Atlanta, GA, pp. 24–35 (September 2002)
4. Pang, Q., Leung, V.C.M., Liew, S.C.: A rate adaptation algorithm for IEEE 802.11 WLANs based on MAC-layer loss differentiation. In: BroadNets. Proceedings of the Seconnd International Conference on Broadband Networks, Boston, MA, pp. 659–667 (October 2005)
5. Kim, J., Kim, S., Choi, S., Qiao, D.: CARA: Collision-aware rate adaptation for IEEE 802.11 WLANs. In: Proceedings of INFOCOM, Barcelona, Spain, pp. 146–157 (April 2006)
6. Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications – 802.11. Institute of Electronic and Electrical Engineers (IEEE) (1999)
7. Braswell, B., McEachen, J.: Modeling data rate agility in the IEEE 802.11a WLAN protocol. In: Proceedings of OPNETWORK (2001)
8. Qiao, D., Choi, S., Shin, K.G.: Goodput analysis and link adaptation for IEEE 802.11a wireless LANs. IEEE Transactions on Mobile Computing 1(4), 278–292 (2002)

9. del Prado Pavon, J., Choi, S.: Link adaptation strategy for IEEE 802.11 WLAN via received signal strength measurement. In: Proceedings of ICC, pp. 1108–1113 (May 2003)
10. Ji, Z., Yang, Y., Zhou, J., Takai, M., Bagrodia, R.: Exploiting medium access diversity in rate adaptive wirless LANs. In: Proceedings of MobiCom, Philadelphia, PA, pp. 345–359 (September 2004)
11. Zhang, J., Chen, Y.P., Marsic, I.: Adaptive MAC scheduling using channel state diversity for wireless networks. In: Proceedings of WiCom (September 2006)
12. Punnroose, R.J., Nikitin, P.N., Stancil, D.D.: Efficient simulation of Ricean fading within a packet simulator. In: Proceedings of VTC, pp. 764–767 (September 2000)

MP-FMIPv6: MLD Proxy Based Fast Multicast Protocol for Mobile IPv6*

Jianfeng Guan, Huachun Zhou, Wei Qu, and Yongliang Hao

Beijing Jiaotong University, 100044 Beijing, China
guanjian863@163.com,hchzhou@bjtu.edu.cn

Abstract. Mobile multicast is a research hotspot and can provide many applications. Some schemes have been proposed to support the mobile subscriber, but most of them study the construction algorithm of dynamic multicast delivery structure and use the analysis and simulation to evaluate the performance, little concern on the multicast disruption time and protocol cost. In this paper, we propose a Multicast Listener Discovery (MLD) Proxy based Fast Multicast protocol for Mobile IPv6 (MP-FMIPv6), and implement simplified MLD proxy function on Home Agent (HA) and extend MLD host part function on MN. The analytical results show that MLD proxy based fast multicast can reduce multicast disruption time and cost. To evaluate the performance of proposed scheme in real applications, we setup a test-bed and compare it with Bi-directional Tunneling (BT) and Remote Subscription (RS) methods. The experiment results show that the proposed scheme nearly improves the multicast disruption time by 2.4 times and 3.66 times, and saves the cost about 19.1% and 57.4%, respectively.

1 Introduction

With the development of mobile and wireless communication technologies, multicast has developed from fixed platform to wireless and mobile platform recently. Mobile multicast has to handle the dynamic membership and host mobility. MLD [1] protocol is used to manage the dynamic membership for IPv6 (IPv4 uses the IGMP), while it can not maintain the continuous multicast session for mobile listeners. To solve this problem, mobile multicast has to support both link layer handover such as IEEE 802.11 and network layer handover such as Mobile IPv6 [2] and NEMO (NEtwork MObility) [3]. However, we can not simply combine them to provide the mobile multicast services.

Mobile IPv4 [4] describes two basic mobile multicast methods, BT and RS. BT has lower join latency and does not require multicast support in foreign networks, but it introduces additional overheads. While RS is more scalable than BT and has optimal routing, but it has long join latency and needs multicast support in foreign networks. Several improved schemes such as Mobile Multicast

* This paper is supported by the Cultivation Fund of the Key Scientific and Technical Innovation Project of China (No. 706005).

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 244–256, 2007.

Protocol (MoM) [5], Range-Based Mobile Multicast (RBMoM) [6], and Multicast by Multicast Agent Protocol (MMA) [7] have been proposed, and they try to make the tradeoff between BT and RS by different multicast agent selection algorithms. However, most of them focus on reconstruction and maintenance of dynamic multicast delivery structure, while little considers minimizing the packet loss and multicast disruption time.

With the development of MIPv6, Fast Mobile IPv6 (FMIPv6) [8] and Hierarchical Mobile IPv6 (HMIPv6) [9] have been proposed. MIPv6 takes about 2∼3 seconds to complete the handover procedures, which is too long to be used for real-time applications. FMIPv6 configures the New Care-of-Address (NCoA) based on the link layer information in advance to shorten the handover delay. It also combines the tunnel between Previous Access Router (PAR) and New Access Router (NAR) to minimize the packet loss. HMIPv6 introduces a new entity called Mobility Anchor Point (MAP) that acts as a local HA to eliminate the overhead of global handover signaling. Based on MLD, Christophe Jelger and Thomas Noel [10] propose a mobile multicast scheme which combines BT and RS by using the MLD-proxy-capable HA. It extends new type of MLD message called Multicast Listener Hold to notify the HA to hold the multicast states but stop forwarding the packets. When MN moves among the foreign networks, it uses the BT method to get multicast data at first, and then change to RS method. However, it has the out-of-synch problem. Based on FMIPv6, F. Xia and B. Sarikaya [11] propose a FMIPv6 extension for multicast handover, which introduces a new Multicast Group Information Option (MGIO) in Fast Binding Update (FBU) and Handover Initiate (HI) message. To reduce the join delay, PAR transmits multicast group information to NAR through FBU and HI with the MGIO to join the multicast group in advance. However, it can not solve the tunnel convergence problem. Afterwards, Georigios A. Leoleis et al. [12] proposed the FMIPv6 extensions for Multicast handover support with Flow Tunneling and Buffering (M-FMIPv6/FTB), which uses the conditional tunneling of multicast traffic per flow to solve the tunnel convergence. More recently, Dong-Hee Kwon et al. [13] proposed an efficient multicast support scheme for FMIPv6. It introduces new multicast options in mobility header and ICMP header, which contain the multicast group information. NAR setups the multicast states MN interested in advance to reduce the join delay. Besides, it establishes a multicast specific tunnel between PAR and NAR to eliminate the tunnel convergence problem. Based on HMIPv6, Thomas Schmidt and Matthias Waehlisch [14] propose a seamless Multicast in HMIPv6 (M-HMIPv6), which uses the MAP as multicast anchor point and transmits all multicast data through it. Besides, Zhang et al [15] propose a MIPv6 multicast scheme with Dynamic Multicast Agent (DMA), which combines movement based method and distance based method to select new multicast agent dynamically to reduce the handoff frequency.

The described schemes above have the following characters. Firstly, they mostly use the simulations to evaluate the performance and lack experiments in real application environment. Secondly, some schemes have to modify or extend the exist protocols. Thirdly, most schemes assume that HA or AR has multicast function which is difficult to deploy. MIPv6 suggests that HA should have a full IPv6 multicast function or proxy MLD [16] application that can provide the same functions with kernel multicast forwarding. So, in this paper we propose MP-FMIPv6 scheme and evaluate its performance in our test-bed. MP-FMIPv6 implements simplified MLD proxy function to forward the MLD messages between multicast router and mobile listeners, and it does not modify the existent mobile support specifications. In our scheme, HA gets the MN information through the binding cache and records multicast group information for them. MN extends MLD host part function to send unsolicited MLD report message once CoA changed. MP-FMIPv6 can greatly simplify the design and implementation of mobile multicast, and removes the complicated multicast function to reduce not only the cost but also the operational overhead of HA. Besides, it is independent of multicast routing protocol in the core networks.

The rest of the paper is organized as follows. Section 2 describes the MP-FMIPv6 scheme in detail. Section 3 analyses and compares multicast disruption time and cost. Section 4 presents the performance evaluation of MP-FMIPv6 in our test-bed. Section 5 gives the conclusions.

2 The Architecture of the Proposed Scheme

MP-FMIPv6 is based on MLD proxy and FMIPv6 to realize mobile multicast. MLD proxy device locates between multicast routers and multicast listeners, and forwards the MLD messages and multicast packets. MLD proxy device must be the Querier in the link to send MLD query message. It can learn and maintain group membership information on behalf of multicast subscribers. MLD proxy configures a single upstream interface attached to multicast router to perform MLD host part function and several downstream interfaces attached to multicast subscribers to perform MLD router part function. It records all subscriptions on the downstream interfaces with record format of (multicast-address, filter-mode, source-list). Based on these records, MLD proxy device forwards the report or leave message from the downstream interfaces to upstream interface, while forward the query message vice versa. In our method, HA implements simplified MLD proxy function and it configures the interface attached to multicast router as the upstream, the bi-directional tunnel as the downstream. When MN moves into another subnet, HA can still maintain the membership information for MN from the tunnel.

2.1 MP-FMIPv6 Function Framework

Figure 1 shows the function architecture of MP-FMIPv6 scheme. HA performs the MLD proxy function and forwards the group membership information and multicast packets between MN and upstream multicast router. HA records themulticast group information and forwarding information for MN. Once HA captures the multicast packets, it will forward them according to the records. When

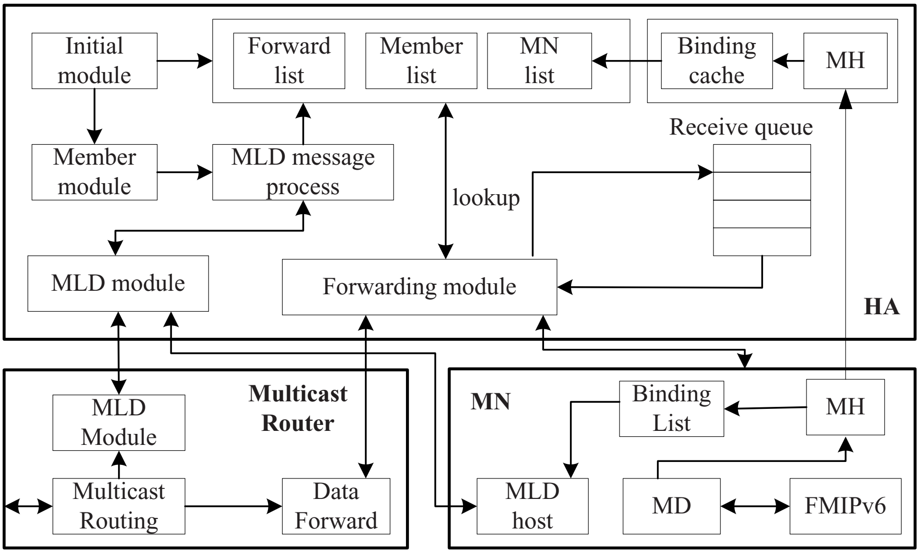


Fig. 1. Function architecture of MP-FMIPv6 scheme

MN moves into another network, it will send an unsolicited MLD report message immediately. PAR and NAR support FMIPv6 and they buffer the multicast data during the handover. Multicast Router attached to HA performs the multicast routing protocol and MLD function.

2.2 MP-FMIPv6 Handover Procedure

Figure 2 shows that MN joins a multicast group in home network and moves into to PAR and NAR. The whole handover procedure consists of four phases.

Phase A: MN at home network. MN sends an unsolicited MLD report message to join a multicast group and get the multicast packets. In this phase, MN joins and leaves the groups like a fixed node.

Phase B: MN moves into PAR. When MN moves into PAR, it performs MIPv6 handover procedure. After that HA gets the MN information from binding cache and maintains the group membership on behalf of MN. Multicast traffic will disrupt when MN lost its attachment with HA until HA receives new BU. In this phase, MN sends unsolicited MLD report message through the tunnel to update the multicast membership information.

Phase C: MN moves from PAR to NAR. MN will perform the FMIPv6 handover during this phase. MN scans available Access Point (AP) nearby before L2 handover. After that MN generates Router Solicitation for Proxy Advertisement (RtSolPr) message and sends to PAR to resolve AP identifier to subnet router information. PAR will reply a Proxy Router Advertisement (PrRtAdv)

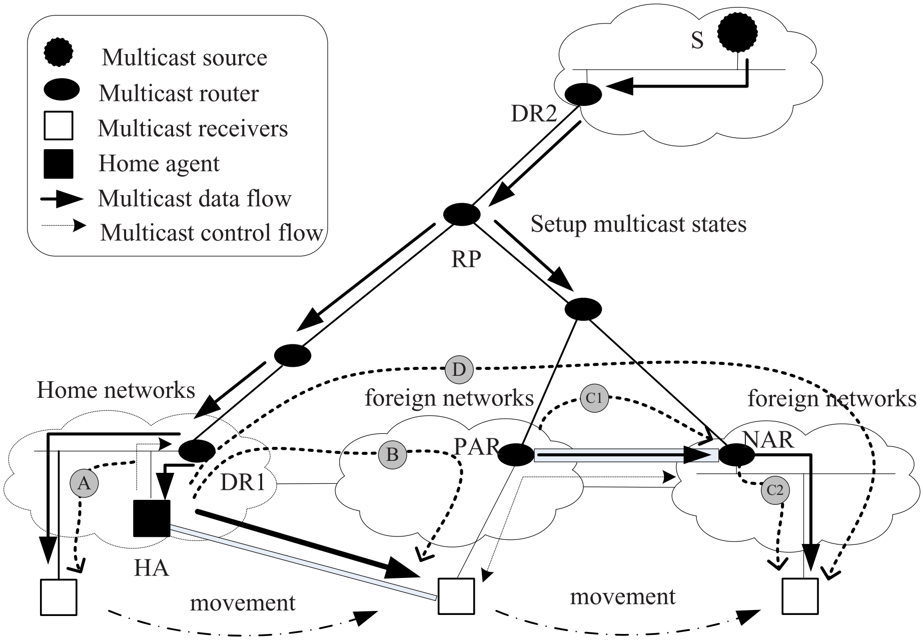


Fig. 2. Operation flow in MP-FMIPv6

message with the [AP-ID, AP-Info] tuple. Multicast traffic will disrupt during scanning procedure. After that MN sends FBU message including NCoA to PAR. Then, PAR and NAR use the HI and Handover Acknowledge (HAck) message to set up a tunnel. After that PAR sends FBack to MN and NAR, and then PAR will buffer and forward the multicast packets to NAR during L2 handover (phase C1). Once MN attaches to NAR and sends FNA message to NAR, it will get the multicast packets from NAR (phase C2).

Phase D: MN attach to NAR. After HA get the new BU from MN, it will modify the tunnel end-point and stops forwarding multicast packets to PAR. As a result, it will transmit to MN directly through NAR.

3 Performance Analysis

Multicast disruption time and cost are two important metrics to evaluate the performance of mobile multicast schemes. In this section, we will analyse and compute them respectively, and compare with BT and RS method.

3.1 Multicast Disruption Time

The multicast disruption time consists of the link layer handover delay and the IP layer handover delay. In this section we use the IEEE 802.11 as the link layer

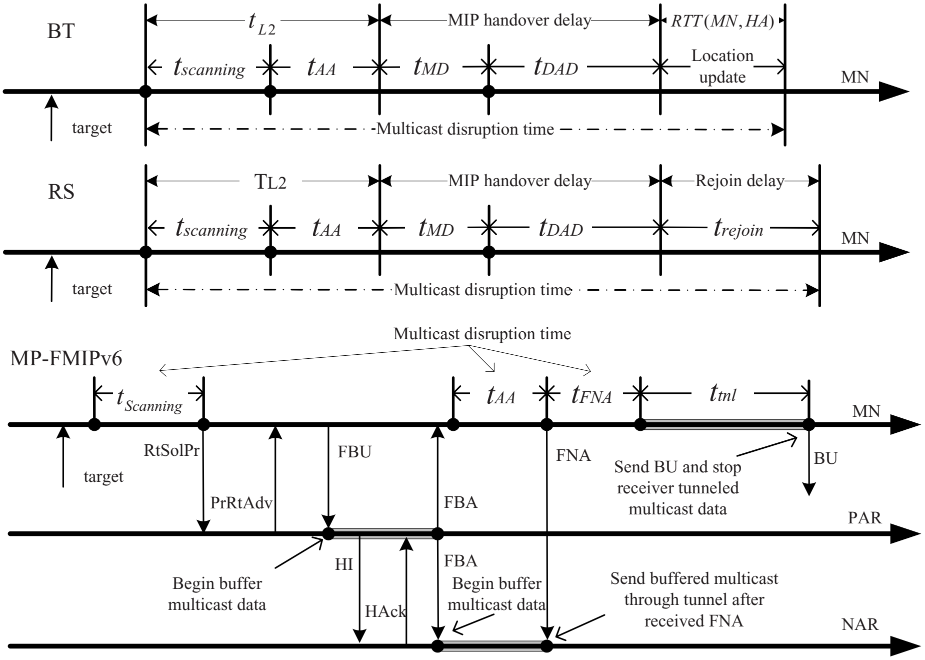


Fig. 3. Comparison of multicast disruption time for BT, RS and MP-FMIPv6

specification. IEEE 802.11 handover consists of scanning precedure ($t_{scanning}$) and authentication and association (t_{AA}) procedure (fig. 3). When Mobile IPv6 is used in IP layer, the disruption time consists of movement detection time (t_{MD}), address configuration time (t_{DAD}), location update with HA and CN (if use route optimization). Movement detection time depends on the sending ratio of RA when stateless address configuration is used. Address configure time depends on DAD process. Locate update is mainly dependent on the route trip time between MN and HA (noted as $RTT(MN, HA)$).

As for the BT method, MN dose not rejoin the group, so the multicast disruption time is

$$t_{BT} = t_{scanning} + t_{AA} + t_{MD} + t_{DAD} + RTT(MN, HA) \tag{1}$$

As for the RS method, MN will surfer from the additional rejoin delay (t_{rejoin}). So the multicast disruption time is

$$t_{RS} = t_{scanning} + t_{AA} + t_{MD} + t_{DAD} + t_{rejoin} \tag{2}$$

When MP-FMIPv6 is used, the disruption time consists of scanning time ($t_{scanning}$) before L2 handover and IP reconnection time (t_{FNA}) after attaching to NAR. However, the L2 handover time only includes the authentication and association (t_{AA}) process because MN has chosen the new AP before L2 handover. So the disruption time of MP-FMIPv6 is

$$t_{MP-FMIPv6} = t_{scanning} + t_{AA} + t_{FNA} \tag{3}$$

3.2 Cost Analysis

Cost consists of signal cost(noted as CS) which is a sum of signal messages and delivery cost (noted as CD) which is the sum of deliver data packets during the handover. In this section, we compute and compare the cost by the hop-based delay analysis method [17]. The parameters are defined as follows and size of signaling messages is shown in table 1.

$C_{BT/RS/MP-FMIPv6}$	Cost of BT/RS/MP-FMIPv6(byte*s)
λ_s	The muticast source rate (packet/s)
λ_r	The sending rate of RA (packet/s)
L_s	The size of multicast packet (byte)
t_{tnl}	The tunnel transmission time (s)
D_{X-Y}	The delivery delay between X and Y (s)
r	The multicast source rate (kb/s)
m	The number of multicast groups MN joined
n	The number of available APs scanned by MN

The total cost of BT is

$$\begin{aligned} C_{BT} &= CS_{BT} + CD_{BT} \\ &= 64 \times (\lfloor \frac{t_{BT}}{\lambda_r} \rfloor + 1) \times D_{MN-AR} + (112 + 16m) \times D_{MN-HA} \\ &\quad + m\lambda_s t_{BT} L_s D_{HA-AR} \end{aligned} \tag{4}$$

Where $\lfloor x \rfloor$ is the maximum integer less than x.

The total cost of RS is

$$\begin{aligned} C_{RS} &= CS_{RS} + CD_{RS} \\ &= 64 \times (\lfloor \frac{t_{RS}}{\lambda_r} \rfloor + 1) \times D_{MN-AR} + 96 \times D_{MN-HA} \\ &\quad + (16 + 16m) \times D_{MN-AR} + m\lambda_s t_{RS} L_s D_{MN-AR} \end{aligned} \tag{5}$$

Table 1. Signaling messages size in byte (IPv6 header is not included)

Message	Used in	size	Message	Used in	size
RS	ICMPv6	16/8	RA	ICMPv6	64
NA	ICMPv6	32/24	BU	MIPv6	56
BA	MIPv6	40	RtSolPr	FMIPv6	24+16n
PrRtAdv	FMIPv6	104	HI	FMIPv6	72
HAck	FMIPv6	32	FBU	FMIPv6	72
FBack	FMIPv6	32/12	MLD Report	MLD	16+16m

The signal cost of MP-FMIPv6 is

$$\begin{aligned} CS_{MP-FMIPv6} = {} & 64 \times (\lfloor \frac{t_{MP-FMIPv6}}{\lambda_r} \rfloor + 1) \times D_{MN-AR} \\ & +(116 \pm 20) \times D_{PAR-NAR} + (112 + 16m) \times D_{MN-HA} \\ & +(212 + 16n \pm 20) \times D_{MN-PAR} \end{aligned} \tag{6}$$

The data delivery cost of MP-FMIPv6 is

$$CD_{MP-FMIPv6} = m\lambda_s L_s t_{tnl}(\frac{40}{L_s} + 1)(D_{PAR-NAR} + D_{NAR-MN}) \tag{7}$$

The total cost of MP-FMIPv6 is

$$C_{MP-FMIPv6} = CS_{MP-FMIPv6} + CD_{MP-FMIPv6} \tag{8}$$

3.3 Numerical Results Analysis

We use the formulas derived from empirical communication delay model [18] and the reference network model shown as figure 4 to compute the delivery delay D_{X-Y}.

$$T_{W-RT}(h, k) = 3.63k + 3.21(h - 1) \tag{9}$$

$$T_{WL-RT}(h, k) = 17.1k \tag{10}$$

Where k is the length of the packet in KB, h is the number of hops, and the T_{TW-RT} and T_{WL-RT} is the round trip time in milliseconds for wired link and wireless link, respectively. To simplify the computation, we define the multicast disruption ratio R_1 and R_2 as following and predefine some values shown in table 2.

$$R_1 = \frac{t_{BT}}{t_{MP-FMIPv6}} \qquad \text{and} \qquad R_2 = \frac{t_{RS}}{t_{MP-FMIPv6}}$$

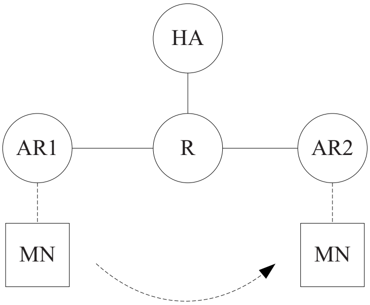


Fig. 4. Reference network model

Table 2. Parameters for numerical analysis

R_1	R_2	$t_{MP-FMIPv6}$	t_{tnl}	$\lambda_s L_s(r)$	λ_r	L_s	n
2/3	3/5	1s	0.8s	32/256kbps	[3,7]	1378byte	5

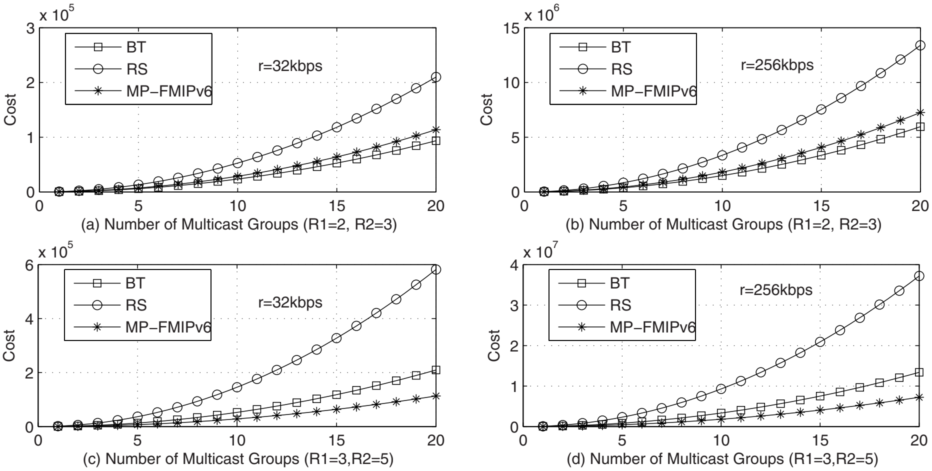


Fig. 5. Cost versus number of groups and multicast source rates for BT, RS and MP-FMIPv6

Assuming that all multicast sources have the same bit rate. Figure 5 shows the cost of the BT, RS and MP-FMIPv6 for different multicast source rates and group numbers. We can get that the cost increases with group number and multicast source rate. Although MP-FMIPv6 has increased many signaling messages compared with BT, its cost is similar to BT and RS, and the difference reduces with an increase of R1 and R2.

4 Performance Evaluation

4.1 Experiments Test-Bed

Figure 6 shows our test-bed topology which consists of three fixed multicast routers, one multicast source, one HA, two access routers, three access points and one MN. The configuration parameter is shown in table 3. Multicast routers and ARs are BJTU star 2600 series which support RIP, RIPNG, OSPF, OSPFv6, BGP, MLDv1/v2 and PIM-SMv2 protocols. HA uses BJTU wireless/mobile router, which supports MIPv6, FMIPv6 and NEMO. Besides, HA implements MP-FMIPv6 to support the mobile multicast. MN is Dell 500 computer with Cisco 350 Aironet series wireless card, and it support MIPv6, FMIPv6, NEMO. Besides, MN implements extended MLD host function which can send unsolicited MLD report message through tunnel and physical interface when MN

Table 3. Parameters for numerical analysis

Message	Default value
Interval of RA	3∼7s
Interval of MLD report	10s
AP send/receiver power	5mw
Wireless card power	1mw
Number of scanned AP	5
Video bit rate	344kb/s
Audio bit rate	32kb/s
Delivery bit rate	632kb/s
Packet length	1378byte

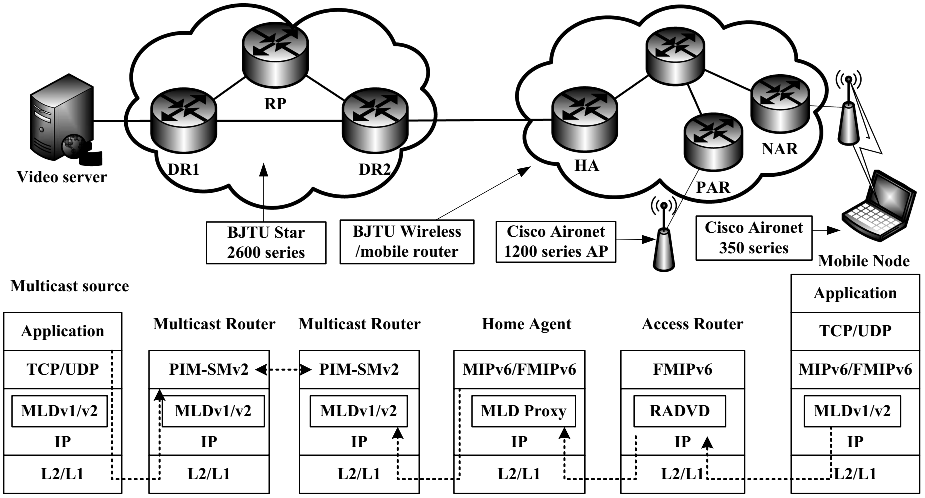


Fig. 6. Test-bed topology

moved. ARs use BJTU star 2600 series router and attach to Cisco Aironet1200 series AP. Multicast source is a PC running VLC player [19] to provide multicast video services. All the network entities in test-bed are running Linux operating system (Fedora Core 2).

4.2 Experiments Results

Multicast Data Flow. We use the Ethereal [20] to capture multicast data of BT, RS and MP-FMIPv6. Figure 7 shows the results of BT and RS.

Figure 8 shows the multicast data flow of MP-FMIPv6. Multicast data in PAR consist of two parts, one part is from 29s to 31.6s when PAR transmitsthe multicast packets to MN, and the other part is form 30.1s to 31.6s when

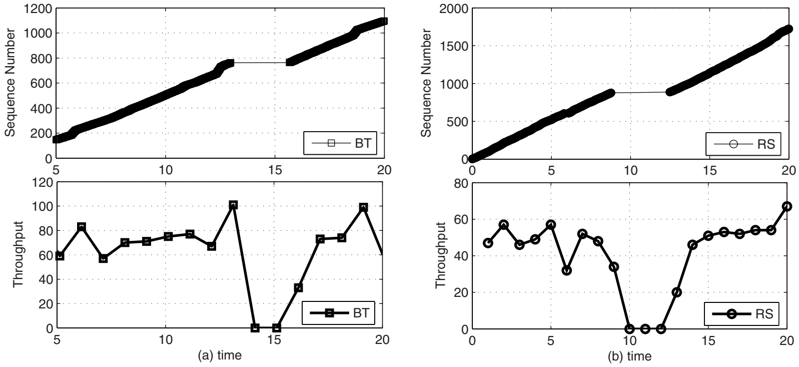


Fig. 7. Multicast data flow of BT and RS

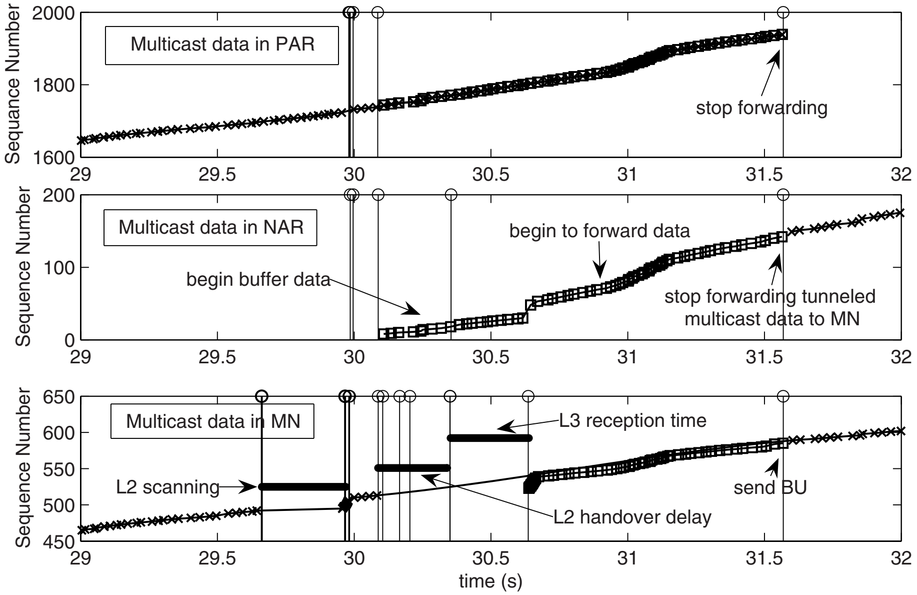


Fig. 8. Multicast data flow in PAR, NAR and MN of MP-FMIPv6 scheme

PAR forwards multicast packets to NAR during handover. After that PAR stops forwarding multicast data. When NAR receives FBack from PAR, it begins to receive the multicast data from PAR, and transmits to MN after receiving FNA. When HA receives the new BU from MN, it will receive the multicast data from HA directly and transmits to MN.

Multicast data in MN has two disruptions during the handover. MN begins scanning AP information in 29.662s and finished scanning in 29.967s. As a result, multicast data disrupted about 305ms, and then it will receive multicast data until MN leaves PAR. At 29.980s MN sends RtSolPr message including five

Table 4. Parameters for numerical analysis)

t_{BT}	t_{RS}	$t_{MP-FMIPv6}$	t_{tnl}
2.678s	3.71s	0.796s	0.9285s

candidate APs to PAR and gets the PrRtAdv message at 29.982s. After that MN configures the NCoA and sends FBU to PAR at 29.983s. After receiving the FBU, PAR sends HI to NAR to perform DAD for NCoA and receives HAck at 30s. At 30.088s PAR sends FBack message to MN and NAR.

MN begins L2 handover at 30.142s and finished at 30.166s, spending about 24ms. After that MN deletes PCoA, updates its route table and configures NCoA at 30.204s. It sends NA message and FNA message to NAR at 30.351s after attaching to NAR. NAR sends buffered multicast packets to MN at 30.633s until MN sends new BU to HA (at 31.569s), and tunneled multicast data transmission lasts about 0.9285s. Multicast data disrupted during L2 and L3 handover (about 491ms). The total multicast data disruption time is about 0.8s.

Multicast Disruption time. From figure 7 and figure 8 we can get the multicast disruption time. BT is 2.687s, and RS is 3.71s, while MP-FMIPv6 is 0.796s (0.305s for scanning and 0.491s for L2 and L3 handover). Compared with the BT and RS, multicast disruption time of MP-FMIPv6 is improved by about 2.4 times and 3.66 times, respectively.

Cost. The experiments parameters are shown in table 4 and the computation results are 6.5563e5 for BT and 1.247e6 for RS, 5.34e4 for MP-FMIPv6, respectively. We can get the MP-FMIPv6 saves the cost by 19.1% against BT and 57.4% against RS.

5 Conclusion

In this paper, we propose and implement a mobile multicast scheme which implements simplified MLD proxy function on HA and extends MLD host part function on MN. We analyze its multicast disruption time and cost, and give out the numerical results which show that the proposed scheme has lower multicast disruption time and cost than BT. Besides, we setup a test-bed to evaluate its performance in real multicast video applications. The experiments results show that the proposed scheme reduces the multicast disruption time and cost. In the future, we will deploy the MLD-proxy function in MAP or AR to improve the performance further.

References

1. Deering, S., Fenner, W., Haberman, B.: Multicast Listener Discovery (MLD) for IPv6, RFC 2710 (1999)
2. Johnson, D., Perkins, C., Arkko, J.: Mobility Support in IPv6, RFC 3775 (2004)
3. Devarapalli, V., Wakikawa, R., Petrescu, A., Thubert, P.: Network Mobility (NEMO) Basic Support Protocol, RFC 3963 (2005)

4. Perkins, C. (ed.): IP mobile support for IPv4, RFC 3344 (2002)
5. Harrison, T.G., Williamson, C.L., Mackrell, W.L., Bunt, R.B.: Mobile multicast (MoM) protocol: Multicast support for mobile hosts. In: ACM MOBIGCOM 1997, Budapest, Hungary, pp. 151–160 (1997)
6. Lin, C.R., Wang, K.M: Mobile multicast support in IP networks. In: IEEE INFOCOM 2000, Tel Aviv, Israel, vol. 3, pp. 1664–1672 (2000)
7. Suh, Y.-J., Shin, H.-S., Kwon, D.-H.: An efficient multicast routing protocol in wireless mobile networks. ACM Wireless Networks 7(5), 443–453 (2001)
8. Koodli, R., et al.: Fast Handover for Mobile IPv6, RFC 4068 (2005)
9. Soliman, H., Flarion, et al.: Hierarchical Mobile IPv6 Mobility Management (HMIPv6), RFC 4140 (2005)
10. Jelger CMulticast for Mobile Hosts in IP NetworksProgress and Challenges. IEEE Wireless Communications 9(5), 58–64 (2002)
11. Xia, F., Sarikaya, B.: FMIPv6 extension for multicast Handover, draft-xia-mipshop-fmip-multicast-01, work in progress (2007)
12. Leoleis, G.A., et al.: Seamless multicast mobility support using fast MIPv6 extensions. Computer Communications 29 (2006)
13. Kwon, D.-H., et al.: Design and Implementation of An Efficient Multicast Support Scheme for FMIPv6. In: IEEE INFOCOMM 2006, Barcelona, Catalunya, Spain, pp. 1–12 (2006)
14. Schmidt, T.C., Waehlisch, M.: Seamless Multicast Handover in a Hierarchical Mobile IPv6 Environment (M-HMIPv6), draft-schmidt-waehlisch-mhmipv6-04, work in progress (2005)
15. zhang, H.-K., Guan, J., et al.: Mobile IPv6 Multicast with Dynamic Multicast Agent, draft-zhang-mipshop-multicast-dma-03, work in progress (2007)
16. Fenner, B., et al.: Internet Group Management Protocol (IGMP) /Multicast Listener Discovery (MLD)-Based Multicast Forwarding (IGMP/MLD Proxying), RFC 4605 (2006)
17. Pfirez, X., Costa, R, Schmitz, H., Hartenstein, et al.: A MIP, FMIPv6 and HMIPv6 Handover Latency Study: Analytical Approach. In: Proc. of IST Mobile & Wireless Telecommunications (submit 2002)
18. Ravi Jain, T., Raleigh, C., Graff, M., Bereschinsky, M.: Mobile Internet Access and QoS Guarantees using Mobile IP and RSVP with Location Registers. In: ICC 1998, pp. 1690–1695 (1998)
19. VideoLan Client (VLC), http://www.videolan.org/vlc
20. Combs, G., et al.: Ehtereal-Network Protocol Analyzer, http://www.ethereal.com

Performance Analysis of a Protocol for Network-Based Localized Mobility Management*

Huachun Zhou

Beijing Jiaotong University, 100044 Beijing, China
hchzhou@bjtu.edu.cn

Abstract. This paper introduces a protocol for network-based localized mobility management solution, and develops an analytic model for the performance analysis based on one-dimensional random walk of mobile node. Based on the analytic models, the location update cost and the packet delivery cost are formulated. Then, the impacts of average cell residence time and the number of mobile nodes in a cell on the total cost are analyzed, respectively. In addition, the variation of the total cost is studied as the session-to-mobility ratio is changed and the optimal local mobility domain size to minimize the total cost is also investigated. The analysis results indicate that the session-to-mobility ratio and the local mobility domain size are critical performance factors in order to minimize the total cost.

1 Introduction

Mobile IPv6 (MIPv6) [1] enable Mobile Node to maintain its connectivity to the Internet during handover. However, mobile IP protocols are mobile node centric that is the handover related decision making is mostly done in the mobile node only. Recently, IETF proposed Network-based Localized Mobility Management (NETLMM) [2] protocol. The NETLMM problem is restricted to providing IP connectivity and reachability for mobile nodes within an access network. A NETLMM scheme requires no localized mobility management support on the mobile node (MN) and is independent of global mobility management protocol. The result is that the mobile node does not have to change its IP address within a restricted access network. However, mobile nodes require global mobility management protocols to support global mobility when moving between different NETLMM access networks [3]. Several related drafts have designed the NETLMM protocols. An IP layer interface between mobile nodes and access routers (ARs) of a NETLMM domain is specified in [4] [5]. By utilizing network-based mobility support for dual stack mobile nodes proposed in [6].

The new NetLMM protocol [7] designed by the NetLMM Design Team is replaced by proxy mobile IP protocols [8] or [9], since, recent developments in network architectures in standards development organizations such as WiMAX

* This research has been supported by the Cultivation Fund of the Key Scientific and Technical Innovation Project of China (No. 706005).

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 257–268, 2007.

forum and 3GPP have identified a need to support Proxy Mobile IP solution. The WiMAX network architecture [10] currently supports Proxy Mobile IPv4 (MIPv4) for enabling mobility for hosts that may not have a MIPv4 client. Proxy MIPv6 is a solution that is aligned with the architectural direction of WiMAX. In 3GPP, there has been some degree of interest in Proxy MIPv6 as well, primarily in the SAE (System Architecture Evolution) [11] work item. The goal to standardize proxy MIPv6 is specify a simple extension to MIPv6 that would support network based mobility for IPv6 hosts and reuses MIPv6 signaling and features.

In this paper, the performance of a protocol for network-based localized mobility management solution [9], named PMIPClient, is analyzed. PMIPClient is a network controlled mobility protocol within a Local Mobility Domain (LMD). PMIPClient is based on Proxy Mobile IPv6 (PMIP) technique and employs a PMIP client that generates a Proxy Binding Update (PBU) message. Specially, PMIPClient allows for a clean separation between the bearer and signaling paths. PMIPClient does not involve change in IP address (Proxy Home Address, pHoA) of the mobile node (MN) as it moves within a LMD.

In the previous NETLMM discussion, HMIP [12] was presented as a candidate solution but considered out of scope since it has host involvement. PMIPClient uses mobile IP derivations, but not use HMIP derivations. [13] proposes a dynamic and distributed HMIP (DDHMIP) scheme. The identity of GFA (Gateway Foreign Agent) is not fixed, and different MN can designate different GFA. [14] proposes a hierarchy structure of chain FA for every MN (DHMIP). When MN handoffs between FAs in the regional network, it only updates the new Care-of Address (CoA) to its previous FA. Thus the new FA forms a new location management hierarchical level for the MN. The packets can be intercepted and retunneled along the FA hierarchy till to MN. Once the number of the hierarchy level reaches a dynamical threshold, MN registers with HA and re-chooses GFA. However, DDHMIP and DHMIP are not network-based, and do not allow for the separation between the bearer and signaling paths.

The remainder of the paper is organized as follows. Section 2 describes the protocol for network-based localized mobility management. The analytical mobility model is described in section 3. Section 4 derives the cost functions using the analytical model. The results are presented in section 5. Finally, in section 6, the conclusions and future work are discussed.

2 PMIPClient Method

A protocol for network-based localized mobility management solution (PMIP-Client) is proposed in [9]. This solution is a network controlled mobility protocol within a local mobility domain (LMD). The LMD is an administrative network that contains one Local Mobility Anchor (LMA) or more LMAs and several Mobile Access Gateways (MAGs) within which an MN can maintain its IP address.

This solution is based on Proxy Mobile IPv6 (PMIP) technique and employs a PMIP client (named PMIPclient) that generates a Proxy Binding Update (PBU) message. The PBU is a standard Mobile IPv6 Binding Update (BU) message described in RFC 3775 [1]. Further, the LMA is a standard Mobile IPv6 Home Agent (HA) as defined in RFC 3775 [1] that receiving Proxy Binding Updates (PBU) from the PMIPclient within a LMD. These MAGs may or may not be co-located with base stations or access points that terminate the layer 2 protocol with the MN. The MAG also terminates correspondent node (CN) routed data traffic from the LMA for MNs and forwards traffic from MNs to the LMA.

The solution allows for a clean separation between the bearer and signaling paths. When there are multiple PMIPclients in the LMD, the PMIPclient for an MN can be assigned statically or dynamically chosen by the MAG the MN is currently attached. The PMIPclient can be co-located at one of the MAGs. This MAG, for example can be the one where the MN first acquired its IP address in the LMD. Even in this case the security keys involved in the Mobile IP signaling shall be anchored at a single MAG. During handoffs, only the tunnel end-point will be migrated to the new MAG, whereas the PBU message will still be generated at the anchored MAG. The security keys shall not be transferred across MAGs during the handoff. Because the PBUs for a given MN always originate from the same PMIPclient as long as the MN is within the LMD.

The Figure 1 describes the PMIPClient location update and packet delivery model, where a local mobility domain has standalone PMIPclient.

When an MAG realizes that an MN has made an L2 handoff to a cell (Intra-LMD mobility), the MAG triggers the PMIPclient to initiate and send a PBU to the LMA, which will establish a mapping between the MN's current IP address (pHoA) and the MAG's IP address (Proxy Care-of Address, pCoA). Traffic destined for the MN's pHoA is tunneled to pCoA by the LMA. When the MN leaves that LMD, the MN may be Mobile IP capable and may send BU messages for Inter-LMD mobility, i.e., global mobility.

This solution does not involve change in IP address (pHoA) of the mobile node (MN) as it moves within a LMD. The MAGs in an LMD advertise the same prefix to the MN in their router advertisement messages as the MN moves

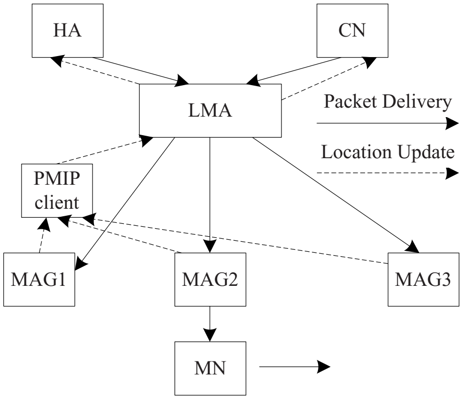


Fig. 1. PMIPClient location update and packet delivery

across different MAGs. So the MN believes that it is still within the same subnet, and thus prevents the MN from initiating Mobile IP messages.

3 Random Walk Mobility Model

This paper incorporates an analytical mobility model [15] to evaluate the performance of the protocol for network-based localized mobility management solution in [9].

This paper assumed the LMD cellular network to have hexagonal cell structure, as shown in Figure 2.

Each LMD is assumed to consist of the same number of rings, K. Each ring $k(k \geq 0)$ is composed of $6k$ cells. So the number of cells up to ring K, N_{MAG}, is calculated as

$$N_{MAG} = \sum_{k=1}^{K} 6k + 1 = 3K(K+1) + 1 \tag{1}$$

The random-walk model is considered as MN mobility model. This random-walk model is one dimensional Markov chain model and is more appropriate for pedestrian movements [16].

In this model, an MN moves to another cell area with a probability of $1-q$ and remains in the current cell with probability, q. In the LMD cellular architecture shown in Figure 2, if an MN is located in a cell of ring k ($k > 0$), the probability that a movement will result in an increase $p^{+}(k)$ or decrease $p^{-}(k)$ in the distance from the center cell is given by

$$p^{+}(k) = \frac{1}{3} + \frac{1}{6k} \text{ and } p^{-}(k) = \frac{1}{3} - \frac{1}{6k} \tag{2}$$

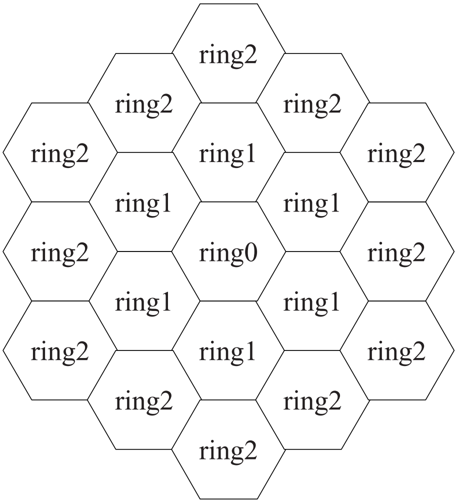


Fig. 2. Cellular configuration in a LMD

The state $k (k \geq 0)$ of a Markov chain is defined as the distance between the current cell of the MN and the center cell. As a result, the MN is said to be in state k if it is currently residing in ring k. The transition probabilities $\alpha_{k,k+1}$ and $\beta_{k,k-1}$ represent the probabilities of the distance of the MN from the center cell increasing or decreasing, respectively. They are given as

$$\alpha_{k,k+1} = \begin{cases} 1-q, & \text{if } \ k=0 \\ (1-q)(\frac{1}{3}+\frac{1}{6k}), & \text{if } \ 1 \leq k \leq K \end{cases} \tag{3}$$

$$\beta_{k,k-1} = (1-q)(\frac{1}{3}-\frac{1}{6k}), \qquad \text{if } \ 1 \leq k \leq K \tag{4}$$

Let $\pi_{k,K}$ be the steady-state probability of state k within a LMD consisting of K rings. Using the transition probabilities in (3) and (4), $\pi_{k,K}$ can be expressed in terms of the steady state probability $\pi_{0,K}$ as

$$\pi_{k,K} = \pi_{0,K} \prod_{i=0}^{k-1} \frac{\alpha_{i,i+1}}{\beta_{i+1,i}} \qquad \text{for } \ 1 \leq k \leq K \tag{5}$$

According to the Markov chain property, the summation of all steady-state probabilities equals to 1. So, $\pi_{0,K}$ can be expressed by

$$\pi_{0,K} = \frac{1}{1+\sum\limits_{k=1}^{K}\sum\limits_{i=0}^{k-1}\frac{\alpha_{i,i+1}}{\beta_{i+1,i}}} \tag{6}$$

4 Cost Functions

Since PMIPClient does not support paging functions, the total cost is the sum of location update cost and packet delivery cost. The location update cost and the packet delivery cost are denoted by CL and CP, respectively. Then, the total cost C_{ost} can be expressed as follows:

$$C_{ost} = CL + CP \tag{7}$$

4.1 Location Update Cost

In PMIPClient solution, an MN performs two types of binding update procedures: the home binding update and the local binding update. The home binding update is a procedure in which an MN registers its pHoA with the HA and CNs. On the other hand, if an MN changes its current address within a LMD, it only needs to register the new address pCoA with the LMA. This registration is referred as a local binding update. CL_H and CL_L denote the signaling costs in the home binding update and the local binding update, respectively. In the IP networks, the signaling cost is proportional to the hop distance of two network entities. CL_H and CL_L can be expressed by the below equations.

$$\begin{aligned} CL_H = {} & 2\eta(D_{MAG-PMIP} + D_{PMIP-LMA} + D_{LMA-HA}) \\ & +2\eta \cdot N_{CN}(D_{MAG-PMIP} + D_{PMIP-LMA} + D_{LMA-CN}) \\ & +CB_{MAG} + CB_{PMIP} + CB_{LMA} + CB_{HA} + N_{CN} \cdot CB_{CN} \qquad (8) \end{aligned}$$

$$CL_L = 2\eta(D_{MAG-PMIP} + D_{PMIP-LMA}) + CB_{MAG} + CB_{PMIP} + CB_{LMA} \qquad (9)$$

where μ and η are the unit transmission costs in a wireless and a wired link, respectively. $D_{MAG-PMIP}$, $D_{PMIP-LMA}$, $D_{MAG-LMA}$, D_{LMA-HA} and D_{LMA-CN} are the hop distance between the MAG and the PMIPclient, the PMIPclient and the LMA, the MAG and the LMA, the LMA and the HA and the LMA and the CN, respectively. Let CB_{MAG}, CB_{PMIP}, CB_{LMA}, CB_{HA} and CB_{CN} be the processing costs for binding update procedures at the MAG, the PMIPclient, the LMA, the HA and the CN, respectively. N_{CN} denotes the number of CNs which are communicating with the MN.

If an MN is located in ring K, the boundary ring of a LMD composed of K rings, and performs a movement from ring K to ring $K+1$, the MN then performs the home binding update procedure. According to the random mobility model proposed in section 3, the probability that an MN performs a home binding update is as follows:

$$\pi_{K,K}\alpha_{K,K+1} \qquad (10)$$

In other cases, except this movement, the MN only performs a local binding update procedure. Hence, the location update cost per unit time can be written as follows:

$$CL = \frac{\pi_{K,K}\alpha_{K,K+1}CL_H + (1 - \pi_{K,K}\alpha_{K,K+1})CL_L}{\overline{T}} \qquad (11)$$

Where $\overline{T}$ is the MN's average cell residence time.

In MIPv6, the signaling costs in the home binding update and the location update cost per unit time can be written by the below equations.

$$\begin{aligned} CL_H = {} & 2[\mu + \eta(D_{MAG-LMA} + D_{LMA-HA})] + CB_{HA} + N_{CN} \cdot CB_{CN} \\ & +2N_{CN}[\mu + \eta(D_{MAG-LMA} + D_{LMA-CN})] \qquad (12) \end{aligned}$$

$$CL = \frac{(1-q)CL_H}{\overline{T}} \qquad (13)$$

4.2 Packet Delivery Cost

In PMIPClient solution, the packet delivery cost can be given as the sum of processing costs for packet delivery at the LMA and the HA (CP_{LMA} and CP_{HA}) and the packet transmission cost from the CN to the MN (CP_{CN}).

In PMIPClient, the processing cost at the LMA is divided into the lookup cost and the routing cost. A LMA maintains a mapping table for translation between pHoA and pCoA. It is used to track the current locations (i.e., pCoA) of the MNs. All packets directed to the MN will be received by the LMA and tunneled to the MN's pCoA using the mapping table. Therefore, the lookup time required for the mapping table needs to be considered. The lookup cost is proportional to the size of the mapping table. The size of the mapping table is proportional to the number of MNs located in the coverage of a LMD. On the other hand, when a packet arrives at the LMA, the LMA selects the current pCoA of the destination MN from the mapping table and the packet is then routed to the MN. Using the longest prefix matching [17], the routing cost is proportional to the logarithm of the number of MAGs belonging to a particular LMD. Therefore, the processing cost at the LMA is given by:

$$CP_{LMA} = \lambda_s \overline{S}[\alpha N_{MAG} N_{CELL} + \beta log(N_{MAG})] \tag{14}$$

In Eq.(14), N_{CELL} denotes the average number of MNs located in the area of an MAG, thus the total number of MNs locates in a LMD is $N_{MAG}N_{CELL}$. In Eq.(14), λ_s denotes the session arrival rate and $\overline{S}$ denotes the average session size in the unit of packet. α and β are the weighting factors.

Since PMIPv6Client supports the route optimization, only the first packet of a session transits the HA to detect whether or not an MN moves into foreign networks. Subsequently, all successive packets of the session are directly routed to the MN. The processing cost at the HA can be written as follows:

$$CP_{HA} = \lambda_s \theta_{HA} \tag{15}$$

In Eq.(15), θ_{HA} refers to a unit packet processing cost at the HA.

The transmission cost in the CN can be expressed by:

$$\begin{aligned} CP_{CN} = \eta\lambda_s[(\overline{S} - 1)(D_{MAG-LMA} + D_{LMA-CN})] + \mu\lambda_s\overline{S} \\ +\eta\lambda_s(D_{MAG-LMA} + D_{LMA-HA} + D_{HA-CN}) \end{aligned} \tag{16}$$

As a result, the packet delivery cost in PMIPClient solution can be obtained as follows:

$$CP = CP_{LMA} + CP_{HA} + CP_{CN} \tag{17}$$

In MIPv6 with Route Optimization, the packet delivery cost can be obtained as follows:

$$\begin{aligned} CP = \eta\lambda_s[(\overline{S} - 1)(D_{MAG-LMA} + D_{LMA-CN})] + \mu\lambda_s\overline{S} \\ +\lambda_s\theta_{HA} + \eta\lambda_s(D_{MAG-LMA} + D_{LMA-HA} + D_{HA-CN}) \end{aligned} \tag{18}$$

5 Numerical Results

This section gives total cost numerical analysis results based on the developed analytic model. Some parameter values for the analysis were referenced from [15] and [18]. They are shown in Table 1.

Figure 3 shows the variation in the total cost as the average cell residence time ($\overline{T}$) is changed. In this analysis, K and N_{CELL} are equal to 4, 4 respectively, and MIPv6 supports route optimization. As the packet delivery cost does not changed, the total cost is affected by the location update cost. The total cost is approximately

Table 1. Numerical analysis parameters

Parameter	μ	η	α	β	λ_s
Value	2	1	0.1	0.2	0.1
Parameter	$\overline{S}$	θ_{HA}	CB_{MAG}	CB_{PMIP}	CB_{LMA}
Value	10	20	6	6	12
Parameter	CB_{HA}	CB_{CN}	N_{CN}	$D_{MAG-PMIP}$	$D_{PMIP-LMA}$
Value	24	6	3	$\lfloor \frac{K}{2} \rfloor$	2
Parameter	$D_{MAG-LMA}$	D_{LMA-HA}	D_{LMA-CN}	D_{HA-CN}	
Value	2	6	4	6	

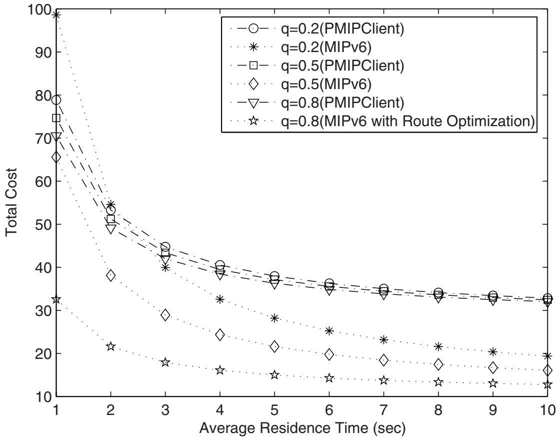


Fig. 3. Total cost vs. average residence time

inversely proportional to the average residence time. The MN performs fewer movements and requires a lower location update cost as the average residence time of an MN increases. In terms of the comparison of PMIPClient and MIPv6 supporting route optimization, MIPv6 has a lower packet delivery cost than PMIPClient. This is because there is no processing cost at the LMA in MIPv6 networks.

Figure 4 shows the impact of the number of MNs in a cell area (N_{CELL}) on the total cost. In this analysis, q and $\overline{T}$ are equal to 0.2 and 5, respectively. As the location update cost does not changed, the total cost is affected by the packet delivery cost. As shown in Figure 4, the total cost increases linearly as the number of MNs increases. In a comparison of PMIPClient and MIPv6, MIPv6 supporting route optimization has a lower total cost than PMIPClient when there is a larger number of MNs in a cell or the size of the LMD is larger. This is because there is no processing cost at the LMA in MIPv6 networks. However, MIPv6 with route optimization has a larger packet delivery cost than PMIPClient when there is a small number of MNs in a cell and the size of the LMD is small. This is because the MN is more likely to perform location update procedure.

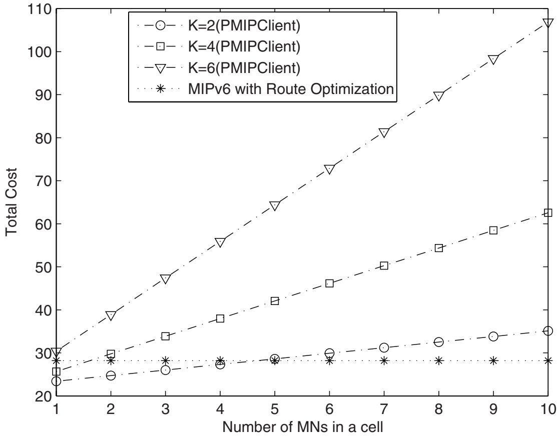


Fig. 4. Total cost vs. number of MNs in a cell

Figure 5 shows the total cost per session ($C_{ost} \cdot \frac{1}{\lambda_s}$) when performance factor called the session-to-mobility ratio (SMR) [15] is changed. The SMR is defined as $\lambda_s\overline{T}$, and it represents the relative ratio of the session arrival rate to the MN mobility rate. Figure 5 shows the total cost when the SMR is 0.1, 1, 10, and 100. In Figure 5, q and N_{CELL} are 0.2 and 4, respectively.

Figure 5 shows the optimal LMD size as the SMR is changed. When the SMR is small, a large LMD size shows a smaller total cost. This is because the location update cost is the dominant factor when mobility rate is larger than the session arrival rate .When the session arrival rate is larger than the mobility rate (i.e., SMR is larger than 1), a large LMD shows a larger total cost due to a higher packet delivery cost.

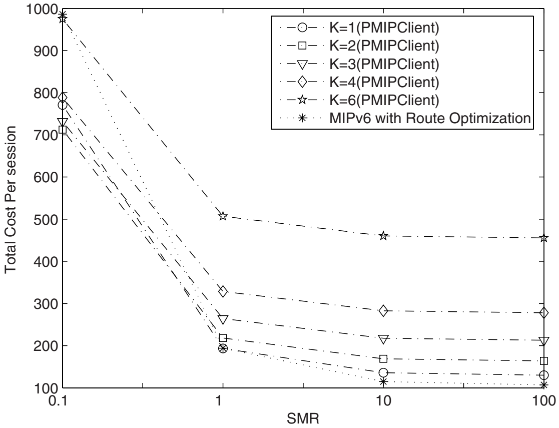


Fig. 5. Total cost per session vs. SMR

In addition, Figure 5 shows MIPv6 requires the largest total cost when the SMR is 0.1. However, when the SMR is larger than 1 (i.e., session arrival rate is higher than the mobility rate), it has the smallest total cost as the packet delivery cost is the dominant factor.

Figure 6 and 7 show the total cost as the LMD size is changed for different MNs. The optimal LMD size is examined as a function of different mobility parameter values of an MN. In Figure 6 and 7, N_{CELL} is 4.

In the case of the static MN, let q and $\overline{T}$ be 0.8 and 8, respectively. Figure 6 shows the location update cost is almost constant for all LMD sizes. This means that the packet delivery cost has an impact on the optimal LMD size. Therefore, the optimal LMD size is 0 when the packet delivery cost is minimized. However,

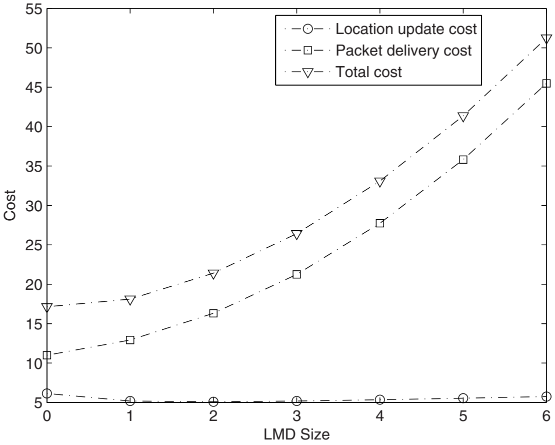


Fig. 6. Total cost vs. LMD size (static MN)

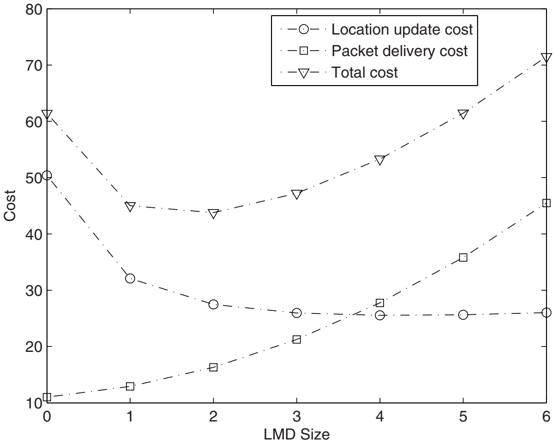


Fig. 7. Total cost vs. LMD size (dynamic MN)

in the case of the dynamic MN, let q and $\overline{T}$ be 0.2 and 2, respectively. Figure 7 shows the location update cost decreases significantly as the LMD size increases. The total cost is at a minimal when the ring size is 2, which is larger than the optimal LMD size of the static MN. Thus, the optimal LMD size increases as the MN mobility rate increases.

6 Conclusion

In this paper, the PMIPClient protocol for network-based localized mobility management solution is analyzed. The location update cost and packet delivery cost was modeled using the random walk model, and the total cost is given. The impacts of cell residence time and the number of mobile nodes in a cell on the total cost, respectively, are analyzed. In addition, the variation in the total cost was examined as the SMR and the LMD size are changed. The analysis results indicate that the SMR and the LMD size are critical performance factors in order to minimize the total cost. The performance of PMIPClient solution is similar to HMIPv6 and it can also minimize the mobility-related signaling costs [15].

Based on the results above, some questions should be further investigated so as to deploy optimal PMIPClient networks. For example, if there are multiple PMIPclients in the domain, the PMIPclient should be chosen dynamically. If there are multiple LMAs, the LMA should be chosen dynamically.

References

1. Johnson, D., et al.: Mobility Support in IPv6, RFC 3775 (June 2004)
2. Kempf, J., Leung, K., et al.: Problem statement for IP local mobility, draft-ietf-netlmm-nohost-ps-04 (June 2006)
3. Njedjou, E., Riera, J.: Problem Statement for Global IP Mobility Management, draft-njedjou-netlmm-globalmm-ps-01 (May 2006)
4. Laganier, J., Narayanan, S.: Network-based Localized Mobility Management Interface between Mobile Node and Access Router, draft-ietf-netlmm-mn-ar-if-01 (June 2006)
5. Templin, F., Russert, S., Chakeres, I., Yi, S.: Network Localized Mobility Management using DHCP, draft-templin-autoconf-netlmm-dhcp-02 (June 2006)
6. Jeong, S., Han, Y.-H., Kim, H.-J.: Network-based Mobility Support for Dual Stack Nodes, draft-jeong-netlmm-dual-stack-moving-00 (June 2006)
7. Giaretta, G., et al.: (NetLMM Design Team): The NetLMM Protocol, draft-giaretta-netlmm-dt-protocol-02 (October 2006)
8. Gundavelli, S., Leung, K., Devarapalli, V.: Proxy Mobile IPv6, draft-sgundave-mip6-proxymip6-01 (January 2007)
9. Bedekar, A., Singh, A., Kumar, V., Kalyanasundaram, S.: A Protocol for Network-based Localized Mobility Management, draft-singh-netlmm-protocol-01 (February 2007)
10. WiMAX End-to-End Network Systems Architecture, (Stage 2: Architecture Tenets, Reference Model and Reference Points): (Accessed on May 2007), http://www.wimaxforum.org/technology/documents

11. 3GPP, 3GPP system architecture evolution (SAE): Report on technical options and conclusions, 3GPP TR 23.882 0.10.1 (February 2006)
12. Soliman, H., Castelluccia, C., El Malki, K., Bellier, L.: Hierarchical Mobile IPv6 Mobility Management (HMIPv6), RFC 4140 (August 2005)
13. Xie, J., Akyildiz, I.F.: A distributed dynamic regional location management scheme for mobile IP. In: Proc. IEEE INFOCOM, 2002, pp. 1069–1078 (2002)
14. Ma, W.C., Fang, Y.G.: Dynamic hierarchical mobility management strategy for mobile IP networks, Selected Areas in Communications. IEEE Journal 22(4), 664–676 (2004)
15. Pack, S., Choi, Y.: A Study on Performance of Hierarchical Mobile IPv6 in IP-based Cellular Networks. IEICE Trans. Commun. E87-B(3), 462–469 (2004)
16. Akyildiz, I.F., Wang, W.: A dynamic location management scheme for next-generation multitier PCS systems. IEEE Trans. Wireless Commun. 1(1), 178–189 (2002)
17. Lampson, B., Srinivasan, V., Varghese, G.: IP lookups using Multiway and Multicolumn search. IEEE/ACM Transactions on Networking 7(3), 324–334 (1999)
18. Zhang, X., Castellanos, J., Capbell, A.: P-MIP: Paging Extensions for Mobile IP. ACM/Kluwer Mobile Networks and Applications 7(2), 127–141 (2002)

A MAC Protocol with Adaptive Preloads Considering Low Duty-Cycle in WSNs

JeongSeok On[1], JaeHyun Kim[1], Jaiyong Lee[1], Yeonsoo Kim[2], and Hakjin Chong[2]

[1] The Graduate School Yonsei University Electrical & Electronic Engineering,
Seoul, Korea
{onchuck, macross7, jyl}@yonsei.ac.kr
[2] USN Research Department Future Technology Lab, KT
yeskim@kt.co.kr

Abstract. Leading MAC protocols developed for duty-cycled WSNs such as B-MAC employ a long preamble and channel sampling. The long preamble introduces excess latency at each hop and results in excess energy consumption at non-target receivers in ultra low-duty cycled WSNs. In this paper we propose AS-MAC (Asynchronous Sensor MAC), a low power MAC protocol for wireless sensor networks (WSNs). AS-MAC solves these problems by employing a series of preload approach that retains the advantages of low power listening and independent channel sampling schedule. The preload massage includes a destination address and a remaining time until data transmission. Moreover AS-MAC offers an additional advantage such as flexible duty cycle as data rate varies. We demonstrate that AS-MAC is better performance than B-MAC through analysis and evaluation.

1 Introduction

The energy efficiency of a sensor node is the most important research theme in wireless sensor networks (WSNs). The energy consumption of the sensor node can divide into three sections which are a sensor, processor and radio unit. Among them the energy consumption of the radio unit occupies the dominant proportion. A MAC protocol directly effects on the energy consumption of the radio unit. Therefore, several MAC protocols that adapt to WSNs have been developed. One of the primary mechanisms to obtain energy efficiency in several MAC protocols is duty cycling [1][2][3][4]. In this approach, each sensor node periodically cycles between an active state and sleep state. In duty cycling, the expansion of sleep period reduces energy consumption of each node. However per-hop latency is increased.

Leading MAC protocols for duty-cycled WSNs are contention-based MAC protocols, since contention-based MAC protocols are scalable in networks topology and adapt easily to mobility of nodes. In addition, the exact synchronization as TDMA-based MAC protocols doesn't be needed in multi-hop [2].

Contention-based MAC protocols are categorized into synchronous and asynchronous approaches. Synchronous protocols, such as S-MAC [2] and T-MAC [3], share a schedule that specifies when nodes are awake and asleep. Exchanging the schedule information during the beginning of active period in order to communicate is control

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 269–280, 2007.

overhead that consume considerable energy. Asynchronous protocols such as B-MAC [4], and WiseMAC [8], rely on low power listening (LPL), so called channel sampling, to link a sender to a receiver who is duty cycling. Because an asynchronous MAC protocol doesn't need synchronization, the implementation can be simplified and its code size can be reduced. However the long preamble in low power listening exhibits two major disadvantages. One is the unnecessary overhearing that causes excessive energy consumption at non-target receivers, and the other is excessive latency at each hop.

We propose a new approach called Asynchronous Sensor MAC (AS-MAC), which employs low power listening scheme. AS-MAC is designed for target monitor application. AS-MAC in ultra low duty cycle can mitigate the overhearing problem and reduce per-hop latency.

In the following, Section 2 describes related work. Section 3 explains the AS-MAC protocol design in detail. Section 4 will present analysis and performance evaluation. The conclusion will be followed by section 5.

2 Related Works

S-MAC [3] is a representative synchronous MAC protocol. S-MAC is a RTS-CTS based MAC protocol that makes use of synchronization between nodes to allow for duty cycling in sensor networks. At the beginning of the active period, the node exchanges synchronization information with its neighbors to assure that the node and its neighbors wake up concurrently. This schedule is only adhered to locally, resulting in a virtual cluster, which mitigates the need for synchronization of all nodes in networks.

B-MAC [5] is an asynchronous MAC protocol. B-MAC is a CSMA-based technique utilizing a long preamble to achieve low power communication. Each node has an independent schedule. If a node has data, it sends the data packet following with a long preamble that is slightly longer than the sleep period of the receiver. During the active period, a node samples the channel and if a preamble is detected, it remains awake to receive the data. With the long preamble, a sender is assured that at some point during the preamble the receiver will wake up, detect the preamble, and remain awake in order to receive the data. It is low power listening (LPL). The major advantage of LPL is that it minimizes active period when there is no traffic.

3 AS-MAC Protocol Design

We propose a new approach called Asynchronous Sensor MAC (AS-MAC). AS-MAC is designed for target monitor application. In this application sensing data is not generated during most of time. But if a sensor detects target movement, the burst data is generated. In this environment, it is desirable for each node to operate in ultra low duty cycle in order to increase energy efficiency. The burst data should be delivered to one sink node quickly. In this situation, a MAC algorithm has long sampling period and has to accommodate burst traffic. AS-MAC can efficiently support these two conflicting design factors; long sampling period for energy efficiency and burst traffic accommodation. AS-MAC works properly in WSN with dense or sparse node density and with low mobility. Data from all nodes is destined to single sink node in the network. We summarize AS-MAC protocol design as bellows.

- Asynchronous protocol
- Unslotted CSMA technique (carrier sensing)
- Channel sampling (LPL: Low Power Listening)
- Preload and ACK frame
- Overhearing avoidance
- Flexible sampling period

3.1 Packet Format and Parameter Design

1) Preload message: Figure 1 (a) shows the preload message format. The sender transmits the successive preload messages prior to data field, which ensures that all nodes in independent schedules can communicate each other. Preamble is used by receiver nodes for chip and symbol synchronization. Delimiter indicates the start of address and duration field. Address field contains the destination address [6]. Duration field informs the remaining time to the start of data [7].
2) Preloads frame: the preloads frame consists of successive preload messages. The preloads frame size is multiple of 11 byte preload message size (see Figure 1 (b)).
3) Preloads period: the preloads period is the duration of preloads frame. Senders can change it according to data rate (see Figure 1 (c)).
4) Request Sampling Period (RSP): If a node has burst sensing data to deliver to sink node, it requests the next forwarding node to change sampling period from the default value to the value indicated in RSP field. Default sampling period is determined by the network manager.

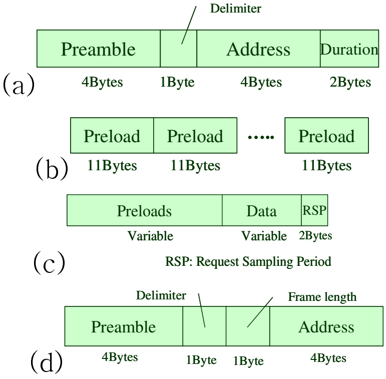


Fig. 1. (a) Preload message format and (b) Preloads frame format and (c) Data frame format, (d) ACK frame forma

Table 1. The preloads table

Preloads period (2bytes)	Address of forwarding node (4bytes)	Timer (2bytes)

Table 2. The sampling table

Sampling period (2bytes)	Address of sender node (4bytes)	Timer (2bytes)

3.2 Operation

We introduce the operation of AS-MAC. This operation is explained by four parts: sender, receiver, overhearing avoidance, flexible sampling period. The sender part represents how each node operates when its buffer is occupied with data. The receiver part describes how each node operates when it detects channel activity during channel sampling. The overhearing avoidance and flexible sampling period parts mitigate two problems.

3.2.1 Sender

When node has the sensing data, carrier sensing is performed to check channel activity (see Figure 2). If the channel is clear, the sender transmits the preloads frame during preloads period (initially, default sampling period). The preloads frame will be followed by data field. The sender transmits the preloads frame as long as default preloads period in order to ensure the forwarding node's wake up. If the sender has burst data packets, it requests the receiver to reduce sampling period using RSP field. RSP field informs the forwarding node of the sampling period to be requested by the sender. How to determine sampling period is mentioned in section 4. The sender should wait an ACK packet from the forwarding node after sending data frame. It tries to retransmit if the ACK packet is missing for the fixed time. If the sender uses RSP field and receives ACK frame from receiver, the sender should reduce preloads period to the duration specified by RSP value and receiver should reduce sampling period according to RSP value. (see Table 1)

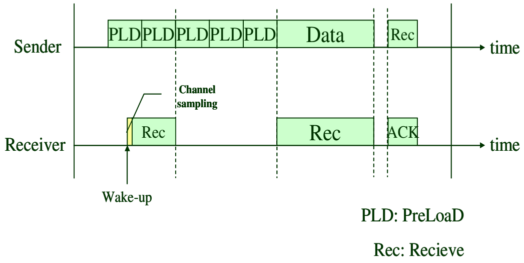


Fig. 2. Operation of AS-MAC

Each node acts as a forwarding node on one or more routing path. A node may be requested to change preloads period by one more nodes. Thus, a node has to maintain a preloads table that includes the preloads period and the next forwarding node. When a node sends data to the next forwarding node, it transmits preloads frame as long as the time defined by the value in the preload table. If the node receives no data during fixed time, the entry of the forwarding node in the preload table is deleted. If the next forwarding node don't be included the preloads table, the node transmits preloads frame during default period

3.2.2 Receiver

If a node (receiver in Figure 2) doesn't have any data to send, it wakes up every sampling period to check channel activity. If another node (sender in Figure 2) is transmitting the preloads frame to send data, the receiver recognizes channel state is active. Entering active mode, the receiver takes the preload message and identifies destination address. If the packet is targeted to oneself, it checks the Duration Value. The receiver can immediately go to sleep mode and keep in this mode during the time indicated in Duration field. The receiver has to wake up after designated duration to receive the preload message from sender. Then the receiver accepts data. The node sends ACK packet to the sender in order to inform successful data packet reception.

If data frame is followed by RSP field, the receiver adds a new entry in the sampling table (see Table 2) to save sender address and sampling period. Then it readjusts its sampling period. Only if newly requested sampling period is shorter than the recently used period, the sampling period is reduced to requested value. That is, a receiver always uses the smallest sampling period in the sampling table. If the node receives no data from the sender during fixed time, the entry of the sender in the sampling table is deleted. Receivers use default sampling period when no entry is existed in the sampling table.

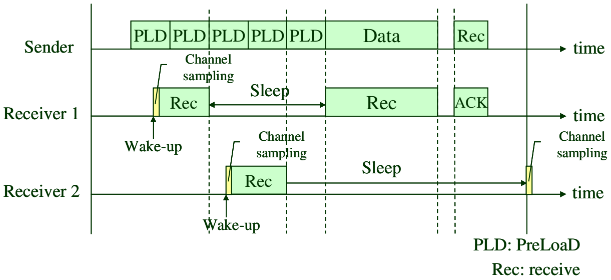


Fig. 3. Overhearing avoidance

3.3 Overhearing Avoidance

In the existing asynchronous schemes, unnecessary energy consumption by overhearing is increased in proportion to sampling period. In AS-MAC, this problem is mitigated by informing of destination address using the transmission of repetitive preload messages. Receivers can identify the destination address of the packet after receiving just a single preload message. And nodes which are not the destination node can immediately enter sleep mode. The node which is designated to the next node can also sleep because it can figure out exactly when data transmission is started. The node wakes up after the time to be indicated duration field in the preload message. Then the node receives the data packet from the sender. Therefore, all nodes in the sender's one hop transmission range save unnecessary energy consumption unlike existing schemes which have to receive long preamble.

Because the energy consumption by overhearing is in proportion to the sampling period, this is serious problem in ultra low duty cycle mode. That is, this makes the energy consumption regular regardless the sampling period. In addition the target node doesn't need to receive all preload messages. This also saves energy consumption by unnecessary hearing (see Figure 3).

3.4 Flexible Sampling Period

In ultra low data rate environment, high energy efficiency can be obtained using long sampling period, so called, ultra low duty cycle (below 0.1%). However one hop delay for packet forwarding is also increased in proportion to sampling period. In this situation, burst traffic can not be quickly delivered in the multi-hop network. AS-MAC solves this latency problem by way that source node requests all forwarding nodes to reduce both channel sampling period and preloads period only when the one has burst data packets to send. The first data packet from source node to sink node may suffer rather large delay. However the following data packets can be forwarded quickly because nodes on the routing path reduce sampling period and preloads period (see Figure 4). This also prevents other nodes from changing the sampling period and preloads period.

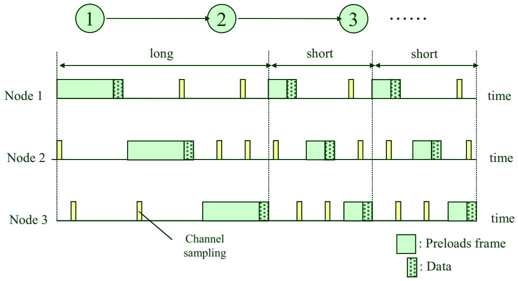


Fig. 4. Flexible sampling period

4 Analysis and Performance Evaluation

We will analyze the expected power consumption of B-MAC (LPL) and AS-MAC. Our analysis is based on a single-hop network model. Consider a network of $n + 1$ nodes, where all nodes can hear each other directly, so each node has n neighbors. Each node generates one data packet every data packet period $1/r_{data}$. Here we consider unicast traffic. Each node is a sender and a receiver once data generation period. Our analysis focuses on the energy consumption by the radio, and we do not consider other components, such as the processor or sensors. There are five radio states: transmitting, receiving, listening, and sleeping. Each state consumes the power of P_{tx}, P_{rx}, P_{listen} and P_{sleep} respectively. Channel sampling is different from normal listening. We denote average sampling duration as $t_{spl,}$ and its average power consumption as P_{sample}. It includes the time that the radio transitions from sleep to listen and the brief sampling time to detect channel activity. We ignore radio transition costs for other states.

Based on the model, we derive the lower bounds of power consumption for both B-MAC (LPL) and AS-MAC.

Both B-MAC (LPL) and AS-MAC are contention-based MACs, so transmission happens after carrier sensing. We denote the average time in carrier sense as t_{csl}. After the date is transmitted, the average time that the sender waits for the ACK packet is denoted as t_{scl}. The energy consumption of the radio is determined by how much time it spends in carrier sensing, transmitting the data frame, waiting for the ACK packet, transmitting/receiving the ACK packet, receiving the data frame, overhearing, sampling channel and sleeping. These are denoted as t_{cs}, t_{tx}, t_{sc}, t_{Ack}, t_{rx}, t_{over}, t_{sample} and t_{sleep} respectively. All these time values are normalized to one second.

Table 3. Symbols used in radio energy analysis, and typical values for the Chipcon CC1000 and CC2500

Symbol	Meaning	CC1000	CC2500
P_{tx}	Power in transmitting	31.2mW	63.6mW
P_{rx}	Power in receiving	22.2mW	38.4mW
P_{listen}	Power in listening	22.2mW	38.4mW
P_{sleep}	Power in sleeping	3uW	1.2uW
P_{sample}	Power in channel sampling	7.4mW	9.6mW
t_{spl}	Avg. time to sample channel	3ms	3ms
t_{csl}	Avg. carrier sense time	7ms	7ms
t_{scl}	Avg. space time	0.2ms	0.2ms
t_B	Time to Tx/Rx a byte	416us	416us
T_P	Channel sampling period	Varying	Varying
r_{data}	Data packet rate	Varying	Varying
L_{data}	Data packet length	50bytes	50bytes
$L_{preload}$	Preload packet length	11bytes	11bytes
$L_{preloads}$	Preloads packet length	Varying	Varying
L_{SP}	sampling period packet length	2bytes	2bytes
L_{Ack}	ACK packet length	10bytes	10bytes
n	Number of neighbors	Varying	Varying

In Table 3, all of our terms are summarized typical values by the Chipcon CC1000 [1] and CC2500 [2]. For both B-MAC (LPL) and AS-MAC, the expected power consumption per node is the sum of the expected power in each state:

$$E = E_{cs} + E_{tx} + E_{rx} + E_{over} + E_{sample} + E_{sleep} \tag{1}$$

We next derive the expected power consumption for both B-MAC (LPL) and AS-MAC.

4.1 B-MAC (LPL) Analysis [5, 11]

B-MAC (LPL) sends a long preamble before each packet. The duration of the preamble is at least the same as the sampling period T_P. The preamble length is

$$L_{preamble} = T_P / t_B \tag{2}$$

where t_B is the time needed to transmit or receive a byte.

The expected power consumption of B-MAC (LPL) is

$$\begin{aligned} E_{LPL} &= E_{cs} + E_{tx} + E_{rx} + E_{over} + E_{sample} + E_{sleep} \\ &= P_{listen} t_{cs} + P_{tx} t_{tx} + P_{rx} t_{rx} + P_{rx} t_{over} \\ &\quad + P_{sample} t_{sample} + P_{sleep} t_{sleep} \end{aligned} \tag{3}$$

The normalized time of each state is demanded in due to (3).

The normalized time of carrier sense is

$$t_{cs} = t_{csl} r_{data} \tag{4}$$

where r_{data} is the data rate on each node. The normalized time of transmitting state is

$$\begin{aligned} t_{tx} &= \left(L_{preamble} + L_{data}\right) t_B r_{data} \\ &= \left(T_P + L_{data} t_B\right) r_{data} \end{aligned} \tag{5}$$

A node will periodically receive n packets from n neighbors. Among them, only one packet is destined for the node. The rest are the overhearing packets. The average time of the received preamble for each packet is $T_P/2$. The normalized time of receiving state is

$$t_{rx} = \left(T_P / 2 + L_{data} t_B\right) r_{data} \tag{6}$$

The normalized time of overhearing state is

$$t_{over} = (n-1)\left(T_P / 2\right) r_{data} \tag{7}$$

The normalized time of channel sampling state is

$$t_{sample} = t_{spl} / T_P \tag{8}$$

The normalized time of sleeping state is

$$t_{sleep} = 1 - t_{cs} - t_{tx} - t_{rx} - t_{over} - t_{sample} \tag{9}$$

Substituting Equations (4)-(9) into (3) and using Equation (2), we obtain the expected power consumption of B-MAC (LPL) as

$$\begin{aligned} E_{LPL} &= \left(P_{tx}\left(T_P + L_{data} t_B\right) + P_{rx}\left(nT_P / 2 + L_{data} t_B + t_{csl}\right)\right) r_{data} \\ &\quad + P_{sleep}\left(1 - \left(t_{csl} + \frac{n+2}{2} T_P + 2 L_{data} t_B\right) r_{data} - t_{spl} / T_P\right) \\ &\quad + P_{sample} t_{spl} / T_P \end{aligned} \tag{10}$$

Where $P_{rx} = P_{listen}$ is used in Equation (10).

Next we turn to the expected power consumption for AS-MAC.

4.2 AS-MAC Analysis

First we derive the expected power consumption in each state from Equation (1). The expected power consumption in carrier sensing is

$$E_{cs} = P_{listen} t_{cs} \tag{11}$$

The energy consumption in transmitting state includes the energy spent for waiting and receiving ACK packet. The expected power consumption in transmitting state is

$$E_{tx} = P_{tx} t_{tx} + P_{listen} t_{sc} + P_{rx} t_{Ack} \tag{12}$$

Where t_{sc} is the waiting time of the ACK packet. The t_{Ack} is the transmitting/receiving time of the ACK packet.

The expected power consumption in receiving state is

$$E_{rx} = P_{rx} t_{rx} + P_{listen} t_{sc} + P_{tx} t_{Ack} \tag{13}$$

The expected power consumption in overhearing state is

$$E_{over} = P_{rx} t_{over} \tag{14}$$

The expected power consumption in channel sampling is

$$E_{sample} = P_{sample} t_{sample} \tag{15}$$

The expected power consumption in sleeping state is

$$E_{sleep} = P_{sleep} t_{sleep} \tag{16}$$

Substituting Equations (11)-(16) into Equation (1) and using equation $P_{rx} = P_{listen}$. The expected power consumption of AS-MAC is

$$\begin{aligned} E_{AS} = {} & P_{rx}\left(t_{cs} + 2t_{sc} + t_{Ack} + t_{rx} + t_{over}\right) + P_{tx}\left(t_{tx} + t_{Ack}\right) \\ & + P_{sample} t_{sample} + P_{sleep} t_{sleep} \end{aligned} \tag{17}$$

Next we will derive the normalized time in each state.

The preloads period is at least the same as the channel sampling period T_P. The length of the preloads frame is $L_{pls} = T_P/t_B$.

The normalized time of carrier sense is equal to Equation (4).

The normalized time of transmitting state is

$$t_{tx} = \left(L_{pls} + L_{data} + L_{sp}\right) t_B r_{data} \tag{18}$$

The normalized time in waiting for ACK packet is

$$t_{sc} = t_{scl} r_{data} \tag{19}$$

The normalized time in transmitting/receiving the ACK packet is

$$t_{Ack} = L_{Ack} t_B r_{data} \tag{20}$$

The average length of the received preload for each data packet is $3L_{pl}/2$. The normalized time of receiving state is

$$t_{rx} = \left(\frac{3}{2}L_{pl} + L_{data} + L_{sp}\right)t_B r_{data} \tag{21}$$

The normalized time of overhearing state is

$$t_{over} = (n-1)\left(\frac{3}{2}L_{pl}\right)t_B r_{data} \tag{22}$$

The normalized time of channel sampling state is equal to Equation (8).

The normalized time of sleeping state is

$$t_{sleep} = 1 - t_{cs} - t_{tx} - t_{rx} - 2t_{sc} - 2t_{Ack} - t_{over} - t_{sample} \tag{23}$$

Substituting Equations (4), (8), (18)-(23) into Equation (17), the expected power consumption of AS-MAC is

$$\begin{aligned} E_{AS} = {} & P_{rx}\left[t_{csl} + 2t_{scl} + \left(\frac{3n}{2}L_{pl} + L_{data} + L_{sp} + L_{Ack}\right)t_B\right]r_{data} \\ & + P_{sample}t_{spl}/T_P + P_{tx}\left[(L_{data} + L_{sp} + L_{Ack})t_B + T_P\right]r_{data} \\ & + P_{sleep}\left\{1 - \left(\frac{3n}{2}L_{pl} + 2L_{data} + 2L_{sp} + 2L_{Ack}\right)t_B r_{data}\right. \\ & \left. - (t_{csl} + 2t_{scl} + T_P)r_{data} - t_{spl}/T_P\right\} \end{aligned} \tag{24}$$

What is the optimal value T_P to minimize the energy consumption, given a fixed n and r_{data}? We can obtain the optimal value by solving the following equation.

$$\frac{dE_{AS}}{dT_P} = 0 \tag{25}$$

Substituting Equation (24) into (25), we find the optimal T_P for AS-MAC is

$$T_P^* = \sqrt{\frac{(P_{sample} - P_{sleep})\cdot t_{spl}}{(P_{tx} - P_{sleep})\cdot r_{data}}} \tag{26}$$

4.3 Performance Evaluation

We compare the energy performance of AS-MAC and B-MAC (LPL) as the sampling period varies. With static traffic loads we can optimize each for maximum energy conservation. Figure 6 shows the power consumption per node in high traffic environment. This result reveals that B-MAC outperforms AS-MAC in low value of the sampling period, since AS-MAC uses the ACK packet for reliable communication. However as the sampling period increasing, the power consumption of B-MAC increases more rapidly than that of AS-MAC.

Figure 7 represents power consumption of AS-MAC and B-MAC (LPL) in low traffic environment. AS-MAC has better performance than B-MAC in all sampling

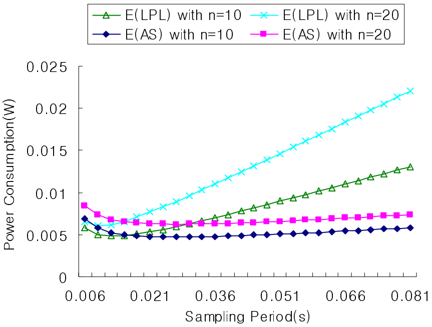
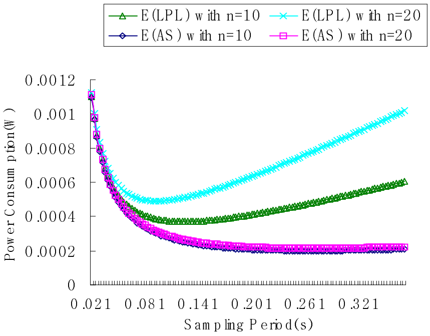


Fig. 6. Mean power consumption of AS-MAC and B-MAC (LPL) with r_{data}=1 and n=10, 20 as the sampling period varies

period. The reason is that the overhearing problem of non-target nodes and the unnecessary energy consumption of the target node are mitigated in AS-MAC. The unnecessary energy consumption of the target node is reduced by duration field of preload message. The energy consumption of the overhearing state is proportional to the sampling period in B-MAC (LPL). However AS-MAC has regular energy consumption of the overhearing state. The overhearing problem is related to the number of neighbors. The energy consumption by overhearing is increased according to the number of neighbors. Therefore, the destination address in preload message prevents the nodes within one hop to overhearing the long preamble.

That is, AS-MAC operates more energy efficient than LPL in ultra low duty-cycled situation (see Figure 8).



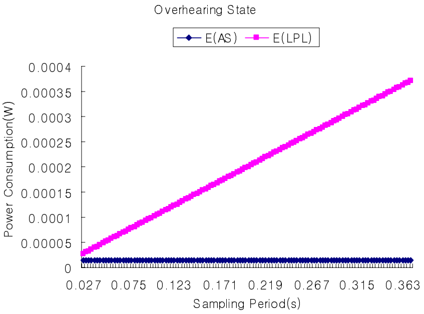


Fig. 7. Mean power consumption of AS-MAC and B-MAC (LPL) with r_{data}=0.01 and n=10, 20 as the sampling period varies (left)

Fig. 8. Energy consumption of overhearing state with r_{data}=0.01 and n=10 as the sampling period varies (right)

5 Conclusion

This paper describes AS-MAC, a new approach for low power communication in WSNs. AS-MAC employs successive preloads approach by transmitting a series of preload messages, each containing the address of the target node and remaining time until data transmission. The series of preload messages prevents non-target nodes from consuming energy in overhearing state. Moreover target node also can sleep during remaining time until data transmission. Then the node will wake for receiving data. This operation can save energy for both non-target nodes and target node.

Flexible preloads and sampling period can allow for lower latency. Nodes on the routing path only change preloads and sampling period. The others don't change preloads and sampling period. In addition AS-MAC finds out the optimal value for preloads and sampling period that minimizes the power consumption per node.

Acknowledgement

This Work was supported by Joint R&D Project ("KT USN Lab @ Yonsei") of KT Corporation, Korea.

References

[1] Chipcon Inc. CC2500 data sheet, http://www.chipcon.com

[2] Chipcon Inc. CC1000 data sheet, http://www.chipcon.com

[3] Ye, W., Heidemann, J., Estrin, D.: Medium access control with coordinated, adaptive sleeping for wireless sensor network. ACM Transactions on Networking 12(3), 493–506 (2004)

[4] van Dam, T., Langendoen, K.: An adaptive energy-efficient mac protocol for wireless sensor networks. In: SenSys. 1st ACM Conference on Embedded Networked Sensor Systems, pp. 171–180 (2003)

[5] Polastre, J., Hill, J., Culler, D.: Versatile low power media access for wireless sensor networks. In: SenSys. The Second ACM Conference on Embedded Networked Sensor Systems, pp. 95–107 (November 2004)

[6] Buettner, M., Yee, G.V., Anderson, E., Han, R.: Media access control: X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks. In: SenSys. Proceedings of the 4th international conference on Embedded networked sensor systems (October 2006)

[7] IEEE, Wireless Medium Access (MAC) and Physical Layer (PHY) specifications for Low Rate Wireless Personal Area Networks (LR-WPANS), IEEE 802.15.4-2003 (2003)

[8] IEEE, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE 802.11, ISO/IEC 8802-11:1999 (1999)

[9] El-Hoiydi, A., Decotignie, J.-D.: Low power downlink mac protocols for infrastructure wireless sensor networks. ACM Mobile Networks and Applications 10(5), 675–690 (2005)

[10] Kim, J.-H., Kim, H.-N., Kim, S.-K., Choi, S.-J., Lee, J.: Advanced MAC Protocol with Energy-Efficiency for Wireless Sensor Networks. In: Kim, C. (ed.) ICOIN 2005. LNCS, vol. 3391, pp. 283–292. Springer, Heidelberg (2005)

[11] Ye, W., Silva, F., Heidemann, J.: Media access control: Ultra-low duty cycle MAC with scheduled channel polling. In: SenSys 2006. Proceedings of the 4th international conference on Embedded networked sensor systems (October 2006)

PTCP: Phase-Divided TCP Congestion Control Scheme in Wireless Sensor Networks

Lujiao Li, Yun Li, Qianbin Chen, and Neng Nie

Special Research Centre for Optical Internet & Wireless
Information Networks, Chongqing University of
Posts & Telecommunications, Chongqing, 400065, China
jiaoer416@gmail.com, {liyun,chenqb,nieneng}@cqupt.edu.cn

Abstract. In wireless sensor networks (WSNs), congestion may occur when sensor nodes are densely distributed and/or burst a mass of data flows, the congestion tends to cause packet loss, which sequentially causes lower throughput and wastes energy. To address this challenge this paper proposes a new congestion control scheme, Phase-divided TCP (PTCP), for wireless sensor networks. It controls the congestion through phase-divided adjusting of the growth rate of the TCP window in the slow-start. Our simulation results demonstrate that the proposed scheme can dramatically resolve congestion control problem, and improve TCP performance of wireless sensor networks.

Keywords: Wireless sensor networks (WSNs); Transport control protocol; growth rate of the TCP window.

1 Introduction

The congestion is likely happen at some nodes in wireless sensor networks because of its limited link bandwidth, small node buffer, many-to-one communication model and multihop routing, which will result in packet dropping, throughput reducing and energy waste. So it is very important to control congestion to prolong network life and improve application quality in WSNs.[1]

Traditional TCP is designed for wired networks of good link quality, and the network congestion is the only reason that causes packet loss. So, once the sender of TCP connection detects the data loss, it runs congestion control scheme and reduces the send window in order to mitigate network load and accordingly avoid network congestion. However, traditional TCP is not suitable for WSNs, the main reasons are detailed as follows [1, 2, 3]: firstly, when packet loss is resulted from link-error, TCP will still trigger congestion control. This style will incur that TCP will unwisely

[1] Supported by the Research Grants by the Science & Tech. Commission of Chongqing (CSTC2006BB2370), the Science and Technology Research Project of Chongqing Municipal Education Commission of China (KJ070521), and Ph.D Grand of Chongqing University of Posts & Telecommunications (A2007-07).

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 281–290, 2007.

reduce data transmission rate under WSNs when there is no congestion. The behaviour will lead to low bandwidth utilization rate, increase network delay and cause lower throughput. Secondly, TCP uses end-to-end approach to control congestion. This approach generally has longer response time when congestion occurs, and in-turn will result in lots of segment dropping. The segment dropping means useless energy consumption and not energy-efficient. Finally, TCP is a connection-oriented protocol. If and only if after the TCP three-way handshake connection has been established, TCP sender can begin to transmit data. In WSNs**,** the topology changes often, frequent TCP connection setup will be a big overhead.

Congestion problem of wireless sensor networks have been brought more and more attentions in academia and industry in recent years, some researchers have been proposed many congestion control algorithms for WSNs. Some well known algorith-ms include CODA [4], ESRT [5], SenTCP [6], STCP [7], Fusion [8], CCF [9] and FeWTCP [10]. CODA and ESRT use buffer occupancy ratio to detect congestion, while SenTCP uses local packet inter-arrival time, service time and buffer occupancy to estimate congestion level; STCP and Fusion use queue length to detect congestion; CCF uses local packet service time to detect congestion; and FeWTCP uses FeW(fractional window increment)scheme to improve TCP performance, which allows the TCP congestion window to grow by α (0<α ≤1) packets at every round-trip-time. Comparing with the above schemes, this scheme will not cause network instability, thus it may avoid congestion. However, when the TCP congestion window (*W*) is small, its growth rate of the TCP window can not increase *W* quickly, it will keep *W* in a small value for a long time, so it can not make use of the network resources effectively. This paper proposes a new congestion control scheme, PTCP (Phase-divided TCP), based on FeWTCP. It uses phase-divided adjusting of the growth rate of TCP window in the slow-start to control the congestion.

2 Related Work

2.1 Change of TCP Window

The TCP-friendly equation is a mathematical model to characterize the steady-state TCP behavior. It describes the TCP window averaged over a long period of time, and shows the relationship between the average window size *W* and the packet loss rate *p*. The literature [10] detailed the average congestion window *W* as Eq. (1),

$$W = \frac{1}{\sqrt{\frac{2p}{3\alpha}} + K\min\{1, 3\sqrt{\frac{3p}{8\alpha}}\}p(1+32p^2)} \tag{1}$$

Where *W* is the average congestion window, *p* is the loss rate, α is the growth rate of TCP window ($\triangle W$) at every *RTT* (round-trip-time), 0<α ≤1, *K=RTO*/*RTT* (*RTO* is retransmission timeout), *K*≥0. We can reduce α and increase *K* to adjust TCP window in light of Eq. (1), Fig. 1[10]shows the relationship between the average window *W* and the packet loss rate *p*, for different α and *K* values. It is obviously to see, the TCP window operational region will drop to a lower range by reducing the value of α from 1 to 0.02. However, by enlarging the value of K from 2 to 50, the decrease of TCP window is small; furthermore, to enlarge K values to such a great one will sharply increase network delay. Thus we conclude that it is more effective to reduce the window growth rate α than increase timeout factor K so as to shift the TCP operational range.

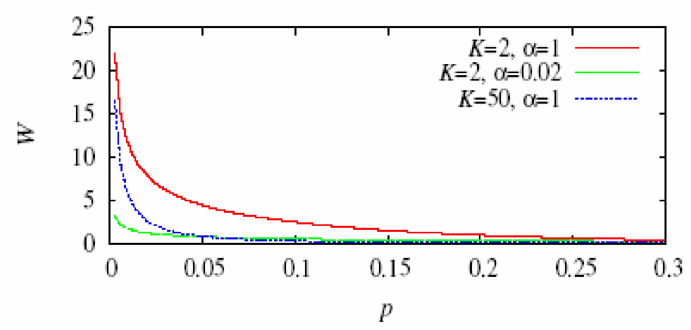


Fig. 1. Comparison of the average window size with three sets of α and K values

2.2 Fractional Window Increment (FeW) Scheme

Based on 2.1, Kitae Nahm proposes a new TCP congestion control scheme called FeWTCP that allows the TCP congestion window to grow by α packets at every *RTT*. This is equivalent to adding of one packet to the window size at every $1/\alpha$ round-trip-time. Suppose that the current TCP congestion window size is *W*, the TCP sender sends *W* packets at every *RTT*, after one *RTT* the TCP sender updates *W* by

$$W^{new} = W^{current} + \alpha ,$$

where $0<\alpha\leq 1$. Because WSNs have a very limited bandwidth, if we increase TCP congestion window by exponential at every *RTT*, this may cause network congestion, the exponential factor is 2. However, in FeWTCP, its window increasing method does not trigger the slow-start of traditional TCP, so it avoids instability when TCP congestion window is exponentially increased in TCP slow-start. Therefore the increasing of TCP congestion window by α packets at every *RTT* will not cause network instability.

3 Proposed Scheme

3.1 Disadvantage of Traditional TCP and FeWTCP

Traditional TCP increases TCP congestion window by exponential, it will cause the transmit data rapid increasing in the network, and in-turn will result in congestion and lots of segment dropping. So it is not suitable for WSNs.

FeWTCP is designed for 802.11 multihop networks of low bandwidth-delay product, which allows TCP congestion window to add α packets at every *RTT*. It will not cause network instability, but it simply grows α packets in spite of TCP congestion window size *W*. When *W* is small, its growth rate of the TCP window $\triangle W$ ($\triangle W = W^{new} - W^{current}$) can not increase *W* quickly, it will keep *W* in a small value for a long time. Thus it can't use network resources efficiently, which results in lower throughput and increases the network delay. And because this scheme keeps $\triangle W$ the same value all the time, it will cause congestion when TCP congestion window reaches to a great value. So it is also not suitable for WSNs.

3.2 Phase-Divided TCP (PTCP) Scheme

Based on 3.1, considering the features of wireless sensor networks in practice, we need a trade-off between FeWTCP and traditional TCP, so that we can increase TCP congestion window by different means in TCP slow-start. It increases the send window quickly so as to use network resources efficiently and reduce network delay when *W* is small in TCP slow-start; and decreases the growth rate of TCP window so as to prevent congestion when *W* is great. Thus we enhance FeWTCP to adjust the growth rate of the TCP window ($\triangle W$) in light of the *W* values, we increase $\triangle W$ while *W* is small and reduce $\triangle W$ while *W* is great (i.e., the growth rate of the TCP window is in inverse proportion to *W*). For convenience, we divide α by *W*, which is equivalent to adding of α / W packets to the TCP window size at every *RTT*.

At the same time, we need to confirm the boundary of *W*, which divides slow-start into two sub-phases. When the *W* is smaller than the boundary, the TCP congestion window increases more than one packet at every *RTT*; in contrast, when the *W* is bigger than the boundary, the TCP congestion window increases smaller than one packet at every *RTT*. We suppose *H* to be this boundary. So we add α / W by (1-α)/(2*H*- *W*), that is we increase TCP congestion window by

$$\frac{\alpha}{W} + \frac{1-\alpha}{2H-W} \tag{2}$$

at every *RTT*.

We will prove why we can increase $\triangle W$ when *W*< *H* and reduce $\triangle W$ when *W*> *H* according to Eq. (2) as follows:

Let

$$y = \frac{\alpha}{W} + \frac{1-\alpha}{2H-W} = \frac{(1-2\alpha)W + 2\alpha H}{W(2H-W)}, \tag{3}$$

$$y_1 = (1-2\alpha)W + 2\alpha H, \tag{4}$$

$$y_2 = W(2H-W) = -(W-H)^2 + H^2 \,. \tag{5}$$

Therefore,

$$y = \frac{y_1}{y_2} \,. \tag{6}$$

As shown in Fig. 2, x-axis denotes *W* values, y-axis denotes *y* values of Eqs. (4) and (5), where y_2 is a parabola, y_1 passes through point (*H*, *H*). From Eq. (4) we can know when $\alpha<0.5$, y_1 is a linear increasing function, and y_1 intersects with y_2 at points 1 and 3 as shown in Fig. 2; when $\alpha>0.5$, y_1 is a linear decreasing function, and the intersections of y_1 and y_2 are points 2 and 4, the relevant *W* of points 1,2,3,4 are 5,6,7,8, respectively. We can know the *W* of points 5 and 6 are smaller than *H* according to Fig. 2 and Eq. (4), and the *W* of points 7 and 8 are bigger than *H*.

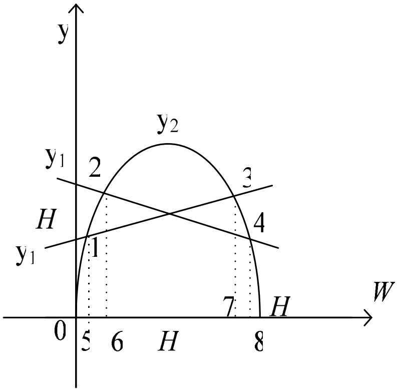


Fig. 2. The relationship between the *y* (growth rate of the TCP window) and the *W*

When the *W* is smaller than the relevant *W* of point 5 (or 6) that is *W*< *H*, from Fig. 2 we can get $y_1>y_2$, thus $y>1$ by Eq. (6). So the $\vartriangle W$ is big comparatively, the TCP congestion window increases more than one packet at every *RTT*. When the *W* is bigger than that of point 5 (or 6), and smaller than that of point 7 (or 8), we can get $y_1<y_2$, thus $y<1$. Thus, the $\vartriangle W$ is small comparatively, the TCP congestion window increases smaller than one packet at every *RTT*. So, according to Eq. (2) we can increase $\vartriangle W$ when *W*< *H* and reduce $\vartriangle W$ when *W*> *H*.

Based on this result, this paper proposes a new flexible TCP window adjusting scheme for wireless sensor networks called PTCP (Phase-divided TCP) that is based on FeWTCP, it carries out as follow:

It allows the TCP congestion window to grow by

$$\frac{\alpha}{W}+\frac{1-\alpha}{2H-W}$$

packets at every *RTT*. To be more precise, the TCP window update can be modified in the following way. Suppose that the current TCP congestion window size is *W*, the TCP sender sends *W* packets at every *RTT*, after one *RTT* the TCP sender updates *W* by

$$W^{new}=W^{current}+\frac{\alpha}{W^{current}}+\frac{1-\alpha}{2H-W^{current}}\ . \tag{7}$$

In this scheme, it can adjust the $\triangle W$ differently with regard of the pre-W value of the network in TCP slow-start. When W is small the $\triangle W$ is big, We know when the W is quite small, increase $\triangle W$ can quickly increase the send window, so it can adequately make use of the bandwidth resources in WSNs and improve throughput. In contrast, when W is great, the $\triangle W$ is small correspondingly. Because wireless sensor networks have a low bandwidth-delay product, when the W is considerably great, reduce $\triangle W$ can effectively prevent network congestion, and thus reduce packet loss. So this scheme can prevent congestion and make best use of the network resources, thus to improve the network throughput.

4 Simulation and Comparisons

We have implemented the proposed scheme using ns-2[11] simulator for 200s. We evaluate and compare PTCP, FeWTCP and TCP over different types of topologies such as chain (in Fig. 3) and 7x7 grid (in Fig. 4)[10]. In the chain topology, node 0 is sender and node *i* is receiver, where *i* is the number of hops. In the 7x7 grid topology, when there is only one TCP flow in networks, node 21 is sender, node 27 is receiver; when there are two TCP flows in networks, node 3 and 21 are sender, node 45 and 27 are receiver. In the simulation, the bandwidth of the wireless channel is 2 Mbps and the radio propagation model is the two-ray ground model with transmission range 250m, carrier sensing range 550m, and interference range 550m, routing protocol is DSR and 802.11 MAC is used.

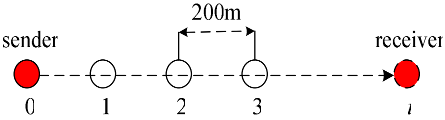


Fig. 3. Chain topology

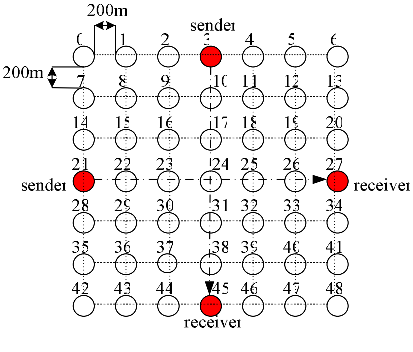


Fig. 4. 7*7 grid topology

4.1 Chain Topology

This paper first compares the throughput of different α when sender sends two TCP flows in the chain topology. In Fig. 5 we show the relationship between the throughput and the α (this paper *H* is adopted half of the slow-start threshold in traditional TCP). We can see that the throughput of PTCP is higher than that of traditional TCP. The throughput of PTCP gradually increases along with reducing of α . When α =0.9, PTCP produces almost the same throughput as traditional TCP, but when α =0.001, the TCP throughput is about 120% more than that of traditional TCP.

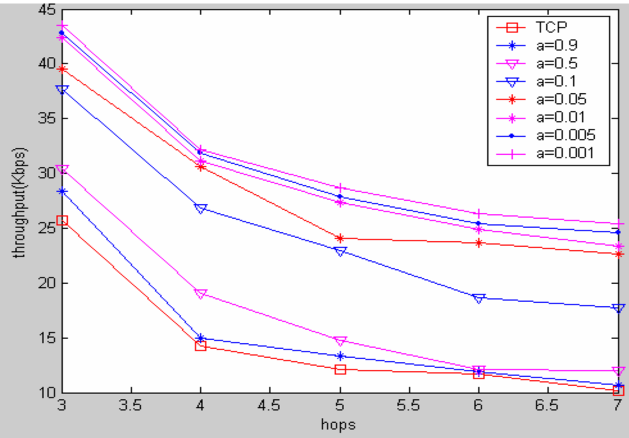


Fig. 5. Throughput comparison of different α values in a chain topology

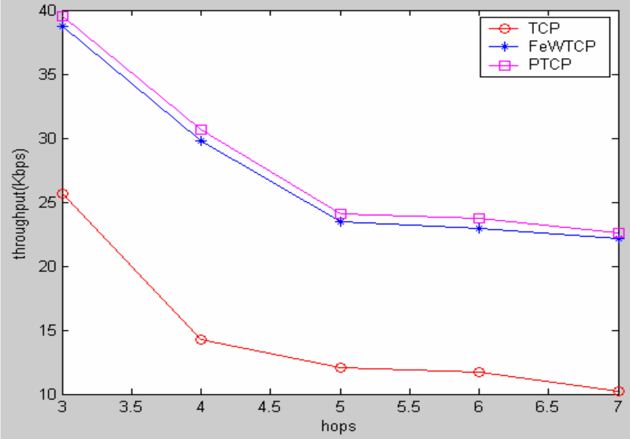


Fig. 6. Throughput comparison of different TCP schemes in a chain topology (2 flows)

Figs. 6 and 7 show that the sender sends two TCP flows and one TCP flow respectively in the chain topology. We compare the throughput of PTCP, FeWTCP

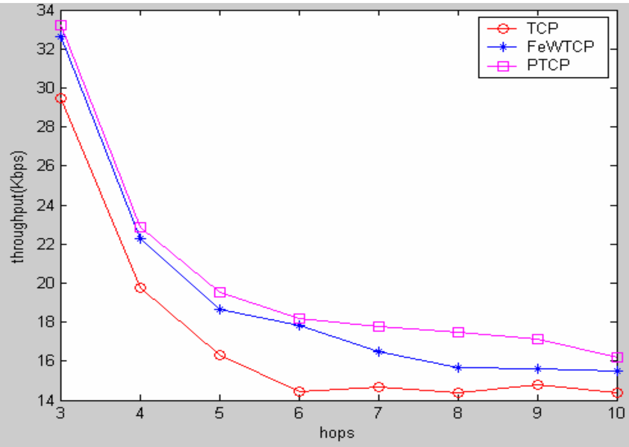


Fig. 7. Throughput comparison of different TCP schemes in a chain topology (1 flow)

and TCP when we increase the number of end-to-end hops in the network. This paper adopts different α values when TCP flow is different, α is chose 0.05 when two TCP flows and 0.5 when one TCP flow. We can see that the throughput of PTCP has a quite obvious improvement compare with that of FeWTCP and TCP when sender sends two TCP flows. It increases about 100% more than that of TCP and about 3% more than that of FeWTCP as shown in Fig. 6. When there is only one TCP flow in the network, the throughput of PTCP has a more obvious improvement, it improves about 10% more than that of FeWTCP as shown in Figs. 7. We can see from Figs. 6 and 7 that the throughput increases gradually as the number of hops increases, that is because as we increase the number of hops, the probability that the network takes place congestion may increase accordingly. The PTCP scheme can phase-divided adjusting of the growth rate of the TCP window in TCP slow-start, so it can prevent the network congestion occurrence and make best use of network resources. Consequently it reduces packet loss and improves the throughput.

4.2 Grid Topology

In Fig. 8 we compare the throughput of these schemes when we increase the number of TCP flows in the 7*7 grid topology, we set α =0.005. We can see that in comparison to FeWTCP and TCP, the PTCP has a gradual improvement as the number of TCP flows increases. That is because as we increase the number of TCP flows, the transmit data increase rapidly in the network, so the probability that the network congestion increases remarkably. The PTCP scheme can phase-divided adjusting of the growth rate of the TCP window in TCP slow-start. So it can effective-ly avoid congestion in the networks. Sequentially it can reduce packet dropping and make use of network resources adequately, so it can improve the network performan-ce effectively.

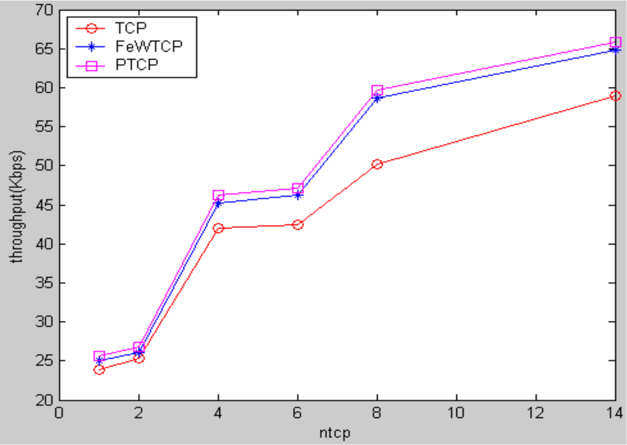


Fig. 8. Throughput comparison of different TCP schemes in a grid topology

5 Conclusions

Aiming to the problem that the congestion control scheme of traditional TCP is not suitable for wireless sensor networks, this paper proposes a new end-to-end congestion control scheme called PTCP (based on FeWTCP) which is based on the growth rate of the TCP window. A series of simulation have proved that the proposed scheme effectively resolve the congestion control problem in WSNs. PTCP scheme has improved the TCP slow-start, which phase-divided adjusting of the growth rate of TCP window according to the TCP congestion window values. So it can prevent congestion and reduce packet loss, it also can make best (comparing to traditional TCP and FeWTCP) use of the network resources and improve the network performance.

References

1. Wang, C., Sohraby, K., Li, B.: Issues of Transport Control Protocols for Wireless Sensor Networks. In: Proc. of 2005 International Conference on Communications, Circuits and Systems, vol. 1, pp. 422–426 (May 2005)
2. Estrin, D., Govindan, R., Heidemann, J., Kumar, S.: Next century challenges: Scalable coordinate in sensor network. In: Proc. of the 5th ACM/IEEE International Conference on Mobile Computing and Networking, pp. 263–270. IEEE Computer Society, Seattle (1999)
3. Wang, C., Sohraby, K., Li, B., Daneshmand, M., Hu, Y.: A survey of transport protocols for wireless sensor networks. IEEE Network 20(3), 34–40 (2006)
4. Wan, C.-Y., Eisenman, S., Campbell, A.: CODA: Congestion detection and avoidance in sensor networks. In: Proc. of ACM Sensys Confs Los Angeles, USA, pp. 266–279 (November 2003)
5. Sankarasubramaniam, Y., Akan, O.B., Akyidiz, I.F.: ESRT: Event-to-Sink reliable transport in wireless sensor networks. In: Proc. of ACM MobiHoc Conf., Anchorage, USA, pp. 177–189 (June 2003)

6. Wang, C., Sohraby, K., Li, B.: SenTCP: A Hop-by-Hop Congestion Control Protocol for wireless sensor networks. In: Proc. of IEEE INFOCOM Conf (Poster Paper), Miami, Florida, USA (March 2005)
7. Iyer, Y.G., Gandham, S., Venkatesan, S.: STCP: A Generic Transport Layer Protocol for Wireless Sensor Networks. In: Proc. IEEE ICCCN Conf., San Diego, CA (October 2005)
8. Hull, B., Jamieson, K., Balakrishnan, H.: Mitigating Congestion in Wireless Sensor Networks. In: Proc. ACM Sensys Conf., Baltimore, MD, pp. 449–454 (November 2004)
9. Ee, C.-T., Bajcsy, R.: Congestion Control and Fairness for Many-to-One Routing in Sensor Networks. In: Proc. ACM Sensys Conf., Baltimore, MD, pp. 148–161 (November 2004)
10. Nahm, K., Helmy, A., Kuo, C.-C.J.: TCP over Multihop 802.11 Networks: Issues and Performance Enhancement. In: Proc. of ACM MobiHoc Conf., Urbana-Champaign, IL, pp. 277–287 (May 2005)
11. Ns-2 simulator, http://www.isi.edu/nsnam/ns/

A Low Latency MAC Scheme for Event-Driven Wireless Sensor Networks

Hung-Cuong Le[1], Hervé Guyennet[1], Violeta Felea[1],
and Noureddine Zerhouni[2]

[1] Laboratoire d'Informatique de l'Université de Franche Comté, France
[2] Laboratoire d'Automatique de Besançon, France
{le, guyennet, felea}@lifc.univ-fcomte.fr
noureddine.zerhouni@ens2m.fr

Abstract. In this paper, we present a low latency media access control scheme which we call LLMAC (Low Latency MAC) for event-driven wireless sensor networks (WSN). In this kind of WSN, sensors do not regularly send data to the sink. They send a burst data only when there is an event in the monitoring area. It takes time for this burst data to arrive to the sink. Normally, these events are critical and we hope to obtain the information on the event in the shortest delay. Hence, the latency is considered to be a crucial requirement in event-driven WSN contrary to the traditional wireless networks where the fairness is the most important requirement. Our proposal LLMAC makes a trade-off between fairness and latency in order to offer a shorter latency transmission when certain events happen. The performance evaluation shows that our proposal reduces the latency in comparison to existing MAC protocols.

1 Introduction

Wireless sensor network is a very hot research topic tendency in distributed systems. A WSN is a network composed of hundreds to thousands of communicating sensors deployed on an area in order to collect environment events. WSN have a wide range of domain application: industry, medical, military, civilization etc. Generally, there are three models of WSN: *continuous*, *on-demand* and *event-driven*. In continuous WSN, sensors send data periodically to the sink. There are always sensors in the network which initiate the communication. In the on-demand WSN model, sensors send data only when they receive a request from the access point. Without request, sensors sense information and store it in their local memory. In the context of this paper, we are interested in the last model: event-driven WSN. In this model, the sensors send data only when certain events occur. For example, a wireless sensor network is deployed on a machine in a factory to detect abnormal symptoms of the machine. Sensors can sense the temperature, the vibration or the humidity of the machine. Normally, when the machine works well, sensors stay silent. When there is a problem in the machine, this problem will produce environment change: an increase in temperature, humidity of the machine or the machine vibrates faster. This ambient information can be detected by sensors and they will activate many camera or sound

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 291–301, 2007.

sensors to take pictures or record sounds and immediately report an event, composed of many packets, to the sink via multi-hop transmission.

In traditional wireless networks, every node is fair in term of channel access. They have the same role in the network. The transmission objective is that when a node has a data to send, it can win the channel in the shortest delay. Hence, almost every existing MAC protocol wants to guarantee the same channel access probability to every node. However, in an event-driven WSN, a communication is often multi-hop and uni-direction from nodes to the sink. An event-driven WSN is often organized in tree topology where each node establishes a data gather tree to send data to the sink. In a tree topology, sensors do not have the same role. Nodes at the leaf do not have to route data. They only have to send data when they detect an event. Nodes in the inner of the network topology have to do two tasks: sense events and route data for other nodes. Hence, in event-driven WSN, sensors are not fair and we need to design a different MAC protocol which is specialized for this kind of networks.

In a WSN, sensors are often spatially correlated. When an event happens, many sensors try to send data to the sink via gateway nodes simultaneously. If every node can transmit data at the same time, there will be no problems. However, in an interfering zone, there is only one node to transmit and all the others have to keep silent. The MAC protocol often guarantees fairness in one-hop communication. Hence, it guarantees that one node can send its packet within the shortest delay in one hop. However, a transmission delay in multi-hop communication is the elapsed time between the moments the packet started to be transmitted and the moment it reaches the sink. Hence, we need to facilitate the channel access of nodes which are nearer to the sink in order to finish the multi-hop transmission within the shortest delay.

In section 2, we present related works of MAC layer of WSNs. Then, we show a simple transmission scenario using IEEE 802.11 where the latency could be a crucial problem. In section 4, we describe LLMAC in order to reduce transmission latency for wireless sensor networks. Next, we show the effectiveness of our approach in comparison with other existing works. Finally, in section 6, we conclude and present several perspectives of our work.

2 Related Works

Today, research on medium access control (MAC) of wireless sensor networks is very fertile. There is a clear attempt to improve MAC protocol management of communication time between sensors, which consumes the most energy. Based on various characteristics, MAC protocol is classified into two different types: *Contention-Free* and *Contention-Based.*

Contention-free MAC is based on reservation and scheduling. Here, each node announces a time slot that it wants to use to the coordinator of the network. This coordinator schedules requests and allocates each node its respective time slot. In this way, a node can access the channel without colliding with others because it is the only node which can transmit during its time slot. Bluetooth [1], TRAMA [2] and LEACH [3] are examples of this type of MAC. This technique guarantees low energy consumption because each node in the network works only during its time slot therefore no collisions. However, the major disadvantage of this technique is that it is not well adaptable to topology changes and is therefore non-scalable. Any insertion or suppression of a node implies a time slot reallocation for all nodes in the group. All contention-free MAC protocols for WSN are designed to support low energy consumption. Hence, they do not take the multi-hop latency into account.

Unlike this technique, *contention-based MAC* is a protocol where every node accesses the channel in competition. Before transmitting a message, a node listens to the channel to see whether there is already a transmission in the medium. If the channel is busy, it waits for a random time and retries to check out the channel later. If the channel is free, it transmits the message.

The most well-known example of this technique is the IEEE 802.11 [4] for wireless LAN network. Indeed, this technique works well in communication between personal computers or pocket PCs, where energy consumption is not a critical problem. However, in a sensor network, the devices are small and very sensitive to energy consumption. Therefore, the MAC technique of IEEE 802.11 is not suitable for sensor networks.

After IEEE 802.11, many research projects have been carried out to optimize the existing MAC protocols to better adapt them to sensor networks. S-MAC [5] is considered to be the first MAC protocol proposal for sensor networks which tries to reduce energy consumption. In S-MAC, nodes are periodically set in listen and sleep mode, where the listen time is approximately 10% of the sleep time. In sleep mode, sensors switch off the radio to save energy. Hence, they can save up to 90% of energy compared to the normal protocols where nodes always stay active. Sensors synchronize their communication during the listen period. If a node does not have any messages to send, it switches its radio off during the sleep mode. On the contrary, it switches its radio on to transmit or receive messages. During listen time, sensors access the channel using the carrier sense multiple access with collision avoidance method (CSMA/CA) [6].

T-MAC [7] extends S-MAC by changing dynamically the listening time between two active periods. T-MAC also reduces the inactive time of the sensors compared to S-MAC. Hence, it is more energy efficient than S-MAC.

B-MAC [8] is a modular and flexible channel access method. The objective of B-MAC is to reduce the idle time of the sensors. Like S-MAC and T-MAC, nodes in B-MAC switch their radio on and off periodically. However, there is no synchronization between sensors. In order for nodes to communicate, packets are sent with a longer preamble than the idle time of sensors.

These MAC protocols are designed for normal WSN where the latency is not a crucial problem. Hence, they do not take the latency into account in their proposal. To the best of our knowledge, SIFT [9] is the first MAC protocol which is designed for event-driven WSN. The main objective of SIFT is to reduce latency in the monitoring application where there are many simultaneous communications. They argue that sensors are spatially correlated. Hence, it is enough for the sink to receive fewer packets for each event. SIFT guarantees successful transmission of R out of N packet (each sensor detects an event and sends one packet) with the shortest delay where N is the number of nodes which detected the event. Based on the same principle, CSMA/p* [10] is proposed where p* is also a non-uniform distribution in order to minimize latency. However, the main objective of this proposal is to reduce the collision but not to reduce the transmission latency for event-driven wireless sensor networks. Moreover, the proposals of SIFT and CSMA/p are applied only for one-hop WSN and each event is composed of one packet. They do not evaluate their propositions in multi-hop scenario and multi-packets per event which is very frequent in an event-driven WSN.

In order to obtain a short latency in certain data transmission scenario, several existing works have been proposed in the domain of QoS in wireless network [11, 12, 13]. The main idea of these proposals is to divide the data flow in different priority levels. By changing the value of the contention window of each priority level, they can favor high priority packets. Hence, high priority packets are sent with the shorter delay than others. Depending on each type of services, packets are assigned a certain priority level. Here, we refer to the differentiated services architecture. These models can be applied only when we know the important level of each type of packets. In case of event-driven WSN, every event is critical and needs to be transmitted to the sink within the shortest delay. We cannot always classify whether one event are more important than the others. Hence, these models cannot be applied directly in event-driven WSN.

3 A Transmission Scenario Using IEEE 802.11

In this section, we present a simple transmission scenario using IEEE 802.11. We will show that IEEE 802.11 can cause latency problems for event-driven WSN.

3.1 The IEEE 802.11 Standard

In this section, we briefly describe the MAC protocol of IEEE 802.11. Fig. 1 illustrates the channel timing of nodes in the network. In fact, the IEEE 802.11 standard proposes two modes: DCF (Distributed Coordinated Function) and PCF (Point Coordination Function). In DCF mode, there is no centralized control. Nodes use CSMA/CA protocol to access the channel in a fully distributed manner. On the contrary, in PCF mode, the base station controls the channel access of all nodes in the network. The MAC protocol is managed locally at the base station. The DCF and PCF mode can work together using different inter-frame spacing value (Fig. 1).

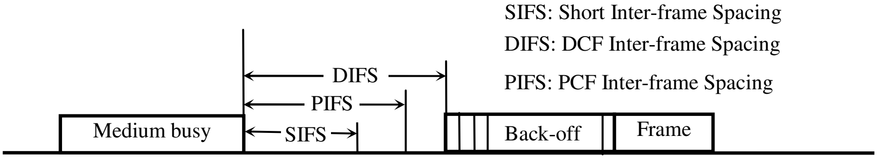

SIFS: Short Inter-frame Spacing
DIFS: DCF Inter-frame Spacing
PIFS: PCF Inter-frame Spacing
DIFS
PIFS
SIFS
Medium busy
Back-off
Frame

Fig. 1. Inter-frame spacing in 802.11

In order to transmit packets, nodes have to wait for a certain time: an inter-frame spacing time and a back-off time (Fig. 1). The IEEE 802.11 standard uses different inter-frame spacing types in order to decide the priority level. The SIFS is reserved for control packets (e.g Request to Send (RTS), Clear to Send (CTS), Acknowledgement (ACK), etc.). These packets have the highest priority and nodes can send packets immediately after the SIFS. The IEEE 802.11 standard gives the PCF mode a higher priority in comparison to the DCF mode. After PIFS, the base station can send scheduling time to nodes in the network without collisions. After SIFS and PIFS expire, all nodes wait till the end of DIFS and start their contending period by choosing a back-off timer in a contention window. The back-off timer is a random number between *[0, CW_{min}]* (the minimal value of contention window). The node which has the shortest back-off timer wins the channel and starts to transmit. All the others have to wait till the end of the transmission. In order to ensure the fairness, all sensors contend the channel after DIFS with the same value of initial contention window. Hence, they have the same probability to access the channel. To prevent a node from occupying the channel for a long time, after transmission of each packet, the node has to release the channel to other nodes. All nodes start to contend the channel again.

3.2 A Simple scenario

Fig. 2 illustrates a simple scenario of an event-driven WSN. In this example, there are four sensors deployed in the network to detect an event. One sensor (the black one) has also the role of gateway to route packets to the sink for other sensors. The hidden terminal problem [6] prevents the network from simultaneous transmissions. If two or more nodes access the channel at the same time, there will be collisions. When an event happens, three sensors detect the event and send this event to the sink via the gateway node. In all existing MAC protocols, to guarantee the fairness in the network, every sensor accesses the channel with the same probability. Hence, we cannot know the transmission order of nodes in the network.

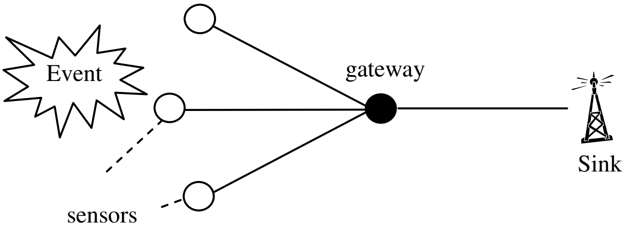


Fig. 2. A simple scenario of an event-driven WSN

As nodes access the channel randomly, we can have different transmission scenarios. Fig. 3 illustrates a pessimistic and an optimistic transmission timeline of the network topology in Fig.2. Each transmission between two nodes takes Δt time. As an event can be composed of many packets: the temperature of the last 10 minutes, a picture of a particular area of a machine, a sound etc, we refer the latency as the elapsed time between the moment that the event happens and the moment that the sink receives all packets of an event. To simplify the illustration, we suppose that each event is composed of two packets.

In the pessimistic scenario (Fig. 3a), as the gateway node is a normal node and it has the same probability to access the channel, it has to wait until all sensors from S1 to S3 finish their transmission. Packets are buffered at the gateway before being transmitted to the sink. Hence, when an event happens, the sink receives the first event after *10*Δt* time. Then, the gateway continues to send packets to the sink and the sink receives all messages of the event after *12*Δt* time.

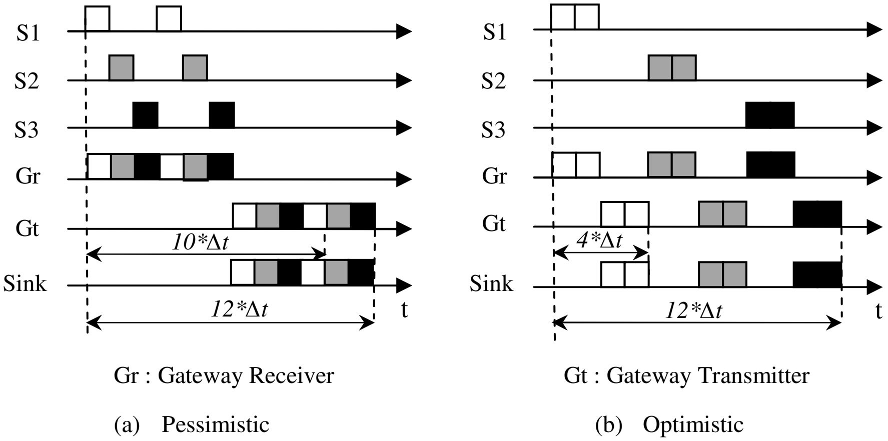


Fig. 3. Transmissions timeline

In the optimistic scenario (Fig. 3b), we have two remarks. First, after sending a packet, the winner sensor continues to occupy the channel and transmits its packet corresponding to the event. It does not release the channel to the others. Second, after receiving a packet from a sensor, the gateway node wins the channel and forwards the packet immediately to the sink. When an event happens, the sink receives the first event after *4*Δt* time. Then, the sink receives all messages related to the event after *12*Δt* time.

As stated earlier, sensors are often spatially correlated and we do not need to receive all messages from all sensors in order to know about the event. The latency of the first R *(R<N)* events is the more important. In the optimistic scenario, the sink receives the first event after *4*Δt* while in the pessimistic scenario, the sink receives the first message after *10*Δt* which is much longer. That is why we call the scenario in Fig.3b an optimistic scenario.

These timelines illustrate just a simple scenario with a two-hop data transmission and two packets for each event. In general cases, the network topology is much larger which implies multi-hop data transmissions. An event can include many packets, it can be a picture taken by an image sensor or a sound recorded by a sound sensor. In these cases, the latency of the optimistic is much lower than the pessimistic scenario.

In all existing MAC protocols, and particularly in IEEE 802.11, nodes access the channel with the same probability. Therefore, the group of three sensors (S1, S2, and S3) has more probability to access the channel than the gateway (three times higher following the theory of probability). The probability that the pessimistic scenario happens is much higher than that of the optimistic scenario. We state that fairness does not optimize the transmission latency, particularly in multi-hop WSN. As we want to reduce the transmission latency, we need to find a method to guarantee that the transmission scenario would follow the optimistic scenario.

4 LLMAC – A Low Latency MAC Protocol for WSN

In this section, we describe our proposal LLMAC which enhances the transmission latency for event-driven WSN. We start by introducing several hypotheses of our working context:

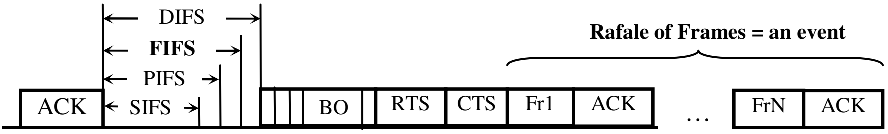
- First, we assume an event-driven WSN to detect certain events in a critical application where the latency is unacceptable.
- Second, each event is composed of a high number of packets: the temperature of the last 10 minutes, a picture, a sound etc.
- Third, we deploy a large WSN where sensors can be organized in a tree-based topology. Hence, when an event happens, sensors send data to the sink via gateway nodes in the data gathering tree [14].
- Finally, sensors are spatially correlated which implies the data redundant transmission. When an event occurs and N nodes detect the event, the sink only needs to receive information from R nodes *(R<N)* in order to have enough information for the event to take a reaction. Hence, the objective of an event-driven WSN is to obtain information from R nodes within the shortest delay.

Starting from these hypotheses, we will present our proposition LLMAC, where we guarantee that the optimistic scenario always happens. We propose two main contributions:

- A change of the transmission policy from frame level to event level
- A new inter-frame spacing value in order to favor the data transmission of gateway nodes

4.1 Rafale of Frame

In normal case, after each frame transmission, nodes have to release the channel and re-contend for another channel access. This prevents the case where a node abuse to occupy the channel all the time. Besides, this method also helps other nodes to have a chance to access the channel. However, this technique is not well adaptable to event-driven where we are interested in the latency of an event, but not that of a frame. Therefore, we change the transmission policy from frame level to event level.



FIFS: Forward Inter-frame Spacing

Fig. 4. Enhancement of IEEE 802.11

Policy 1: *After winning the channel, a node continues to transmit its packets until all packets concerning the event are completely sent.*

In our proposal, we want nodes to complete their transmission for each event before leaving the channel for the others. Hence, in Fig. 4, in place of releasing the channel after each frame transmission, the winner node occupies the channel to send a rafale of frames. This rafale of frames can be a combination of sensed information: temperature, vibration, picture etc. The length of the rafale of frame is equal to the length of an event. In fact, by using a rafale transmission, other nodes lose the right to transmit and we do not guarantee the fairness of the system. However, we see a sensor network as a network with sensors which cooperate to achieve the same goal but not as a network where sensors compete to access the channel with different goals.

As a node can occupy the channel during the transmission of an event, it is not intercepted by other transmission and the latency for an event transmission is reduced. Here, we really make a trade-off between fairness and latency of the event. However, in event-driven WSN, as a short latency is crucial, it is preferable to scarify the fairness in order to obtain better latency results.

4.2 Forward Inter-frame Spacing

By using a rafale of frame transmission, we can guarantee a short delay transmission for each event in one-hop. However, an event-driven WSN is often large with multi-hop transmissions. Moreover, we have shown in the previous section that nodes use the same inter-frame spacing value (DIFS) and the same back-off value to guarantee the fairness. However, the fairness often makes data to be blocked at the gateway node, which increases the transmission latency. Therefore, we need to assure that the forwarding node can win the channel immediately after receiving a packet. Then, data will be forwarded immediately and will not be blocked at the gateway node. Here, we define a new inter-frame spacing that we call FIFS (Forward Inter-frame Spacing) (Fig. 4). This inter-frame spacing is shorter than DIFS and longer than SIFS and PIFS.

Policy 2: *After receiving a packet, a forwarding node accesses the channel immediately with the Forward Inter-frame Spacing.*

We know that inter-frame spacing defines priority level for channel access. By setting the FIFS shorter than DIFS, we give the priority to the transmission of nodes which use the FIFS. As forwarding nodes use this inter-frame spacing, they can access the channel before the contending period of normal nodes. After receiving a rafale of frames for an event, the forwarding node sets its inter-frame spacing to FIFS. Hence, once the FIFS expires, it starts to forward the rafale of frame immediately. This avoids the case where data is blocked at the gateway. The event can be sent multi-hop to the sink within the shortest delay.

5 Performance Evaluation

In this section, we prove the effectiveness of our proposal LLMAC in comparison to IEEE 802.11. As we refer to an event-driven WSN, the most important criterion that we are interested in is the latency of the event.

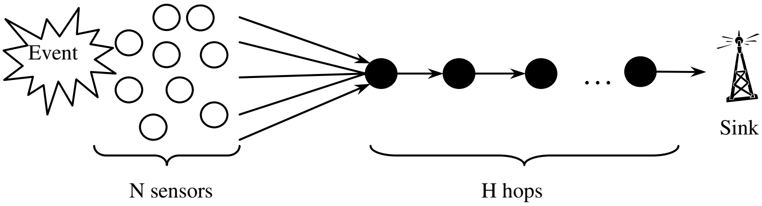


Fig. 5. Simulation topology

We simulate a multi-hop event-driven WSN with the topology described in Fig. 5. We use the OMNet++ simulator [15] to validate LLMAC. OMNet++ is a public-source, component-based, modular and open-architecture simulation environment. When an event happens, N sensors can detect the event and send it to the sink via H hops. Each event is composed of P packets. Each packet has a size equal to the maximal size of an 802.11 frame. All other MAC parameters are set to the standard 802.11. The value of FIFS is set to be an average value between SIFS and DIFS. The objective of an event-driven WSN is to receive R *(R<N)* events within the shortest delay.

5.1 Variation of R

In this simulation, we measure the transmission latency of R events. We set a WSN with these parameters: *N=4, H=2, P=10*. We vary the R value from 1 to N in order to know the latency of each event.

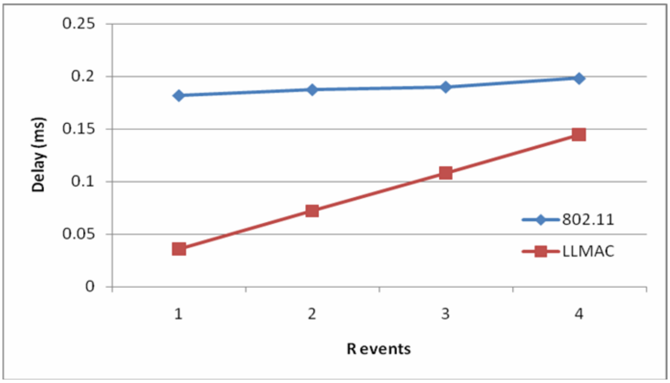


Fig. 6. Latency by varying R

Fig. 6 illustrates the transmission latency of the first R events. There are just four detected events by four nodes. Our proposal always guarantees the shorter delay in comparison to 802.11 in every case. With small R value, LLMAC performs better results in comparison to 802.11 because LLMAC favors the transmission of each event in multi-hop while 802.11 favors the transmission of each packet in one-hop. As R increases, the transmission delay of LLMAC is slightly increased. However, even in the case when all events arrive at the sink, LLMAC always performs a better result than 802.11 because LLMAC uses only one RTS/CTS exchange for all transmission of one event. Between each packet transmission of an event, LLMAC does not use RTS/CTS.

5.2 Variation of P

As mentioned in the previous section, our proposal is aimed to applications where an event is composed of many packets. In this simulation, we want to show the effect of changing the number of packets per event in our proposal and 802.11. We consider a WSN with *N=4, H=2, R=1*. We vary the P value from 10 to 40 packets. In all case, we see that LLMAC always performs a better result than 802.11. When we increase the number of packets for each event, the latency of LLMAC is slightly increased. However, the latency of 802.11 is increased much faster.

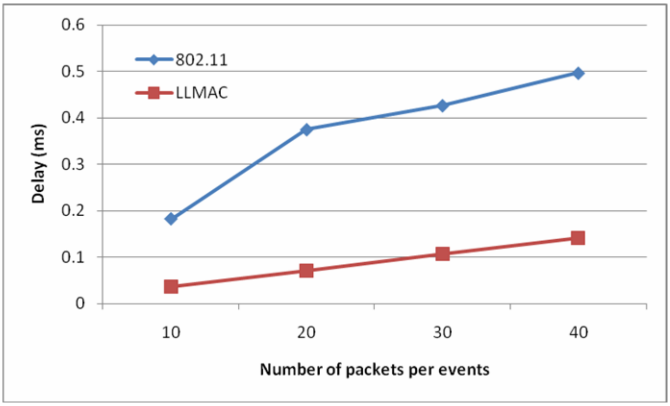


Fig. 7. Latency by varying P

6 Conclusion and Future Works

In this paper, we have presented and analyzed an enhancement of 802.11 which we call LLMAC, a low latency MAC protocol for event-driven wireless sensor network. In this type of WSN, the latency is the most important criterion which decides on the effectiveness of the system. Our method makes a trade-off between fairness and latency in order to offer a shorter latency. By simulation evaluations, we have proved that our proposal clearly improves the latency transmission in comparison to 802.11.

Most of existing works for MAC protocol in WSN often propose a sleep mode for sensors in order to save energy. In sleep mode, nodes cannot sense or transmit information. The latency would be very high in this type of MAC protocols. Therefore, they are more suitable for monitoring application but not for event-driven WSN. Our proposal does not take the energy consumption into account. In the future, we hope to improve the LLMAC with energy consumption awareness. Moreover, we will try to evaluate our proposal in a real testbed framework in the near future.

Acknowledgements

We would like to express our thanks to the Franche Comté region for their financial support for our research at the Franche Comté University, Besançon, FRANCE.

References

1. Specification of the Bluetooth System: Core. (2001) [Online]. Available: http://www.bluetooth.org/
2. Rajendran, V., Obraczka, K., Garcia-Luna-Aceves, J.J.: Energy-Efficient, Collision-Free Medium Access Control for Wireless Sensor Networks. In: Proc. ACM SenSys 2003, Los Angeles, California, pp. 181–192 (November 2003)
3. Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H.: Energy efficient communication protocols for wireless microsensor networks. In: Proc. Hawaii Int. Conf. Systems Sciences, pp. 3005–3014 (January 2000)
4. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification, IEEE Std. 802.11-1999 edition
5. Ye, W., Heidemann, J., Estrin, D.: Medium Access Control With Coordinated Adaptive Sleeping for Wireless Sensor Networks. IEEE/ACM Transactions on Networking 12(3), 493–506 (2004)
6. Tanenbaum, A.: Computer Networks, 4th edn. Prentice Hall, Pearson Education (2003)
7. van Dam, T., Langendoen, K.: An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks. In: ACM SenSys 2003 (November 2003)
8. Polastre, J., Hill, J., Culler, D.: Versatile Low Power Media Access for Wireless Sensor Networks. In: Proc. of the ACM SenSys Conf., Baltimore, pp. 95–107 (2004)
9. Jamieson, K., Balakrishnan, H., Tay, Y.C.: Sift: A MAC protocol for event-driven wireless sensor networks. In: EWSN. Third European Workshop on Wireless Sensor Networks (February 2006)
10. Tay, Y.C., Jamieson, K., Balakrishnan, H.: Collision-minimizing CSMA and Its Applications to Wireless Sensor Networks. IEEE Journal on Selected Areas in Communications 22(6), 1048–1057 (2004)
11. Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., Weiss, W.: An Architecture for Differentiated Services (December 1998)
12. Benveniste, M., Chesson, G., Hoehen, M., Singla, A., Teunissen, H., Wentink, M.: EDCF: Proposed Draft Text. IEEE working document 802.11-01/131r1 (March 2001)
13. Romdhani, L., Ni, Q., Turletti, T.: AEDCF: Enhanced Service Differentiation for IEEE 802.11 Wireless Ad-hoc Networks. In: WCNC2003. The proceeding of the IEEE Wireless Communications and Networking Conference, New Orleans, Louisiana, USA, pp. 1373–1378 (March 2003)
14. Krishnamachari, B., Estrin, D., Wicker, S.: The impact of data aggregation in wireless sensor networks. In: International Workshop on Distributed Event-based Systems (2002)
15. OMNet++ Community Site, [Online] Available http://www.omnetpp.org/

A Location Aided Flooding Protocol for Wireless Ad Hoc Networks

Xinxin Liu[1,2], Xiaohua Jia[3], Hai Liu[3], and Li Feng[2]

[1] School of Computer Science, Wuhan University, Wuhan, China
cslxx@hotmail.com
[2] Wuhan Digital Engineering Research Institute, Wuhan, China
fengli_xjtu@163.com
[3]Dept of Computer Science, City University of Hong Kong, HongKong
{jia@cs, liuhai@cs}cityu.edu.hk

Abstract. Flooding in wireless ad hoc networks is a fundamental and critical operation in supporting various applications and protocols. However, the traditional flooding scheme generates excessive redundant packet retransmissions, causing contention, packet collisions and ultimately wasting precious limited bandwidth and energy. In this paper, we propose an efficient flooding protocol called *vertex forwarding*, which minimizes the flooding traffic by leveraging location information of 1-hop neighbor nodes. Our scheme works as if there were existing a hexagonal grid in the network field to guide the flooding procedure, only the vertex nodes which are located at or nearest to the vertices of the grid should be nominated to forward the message. We also provide a distributed algorithm for finding the vertex nodes. Simulation results show that our scheme is so efficient that it is almost able to reduce the number of forward nodes to the lower bound.

1 Introductions

Flooding is a simple broadcast protocol for delivering a message to all nodes in a network. Many ad hoc network protocols (e.g., routing, service discovery, etc.) use flooding as the basic mechanism to propagate control messages. The simplest flooding technique is called *pure flooding* [1, 2]. In this scheme, every node in the network retransmits the flooding message when it receives it for the first time. Despite of its simplicity, *pure flooding* generates excessive amount of redundant network traffic, because it requires every node to retransmit the message. This will consume a lot of energy resource of mobile nodes and cause the congestion of the network. Furthermore, due to the broadcast nature of radio transmissions, there is a very high probability of signal collisions when all nodes flood the message in the network, which may cause some nodes fail to receive the flooding message. It is so called the *broadcast storm problem* [3]. Sinha et al [4] claimed that "in moderately sparse graphs the expected number of nodes in the network that will receive a broadcast message was shown to be as low as 80%."

To alleviate the broadcast storm problem, several efficient flooding schemes have been presented for ad hoc networks. The most notable works are in [5, 6, 7, 15, 17].

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 302–313, 2007.

However, these algorithms either perform poorly in reducing redundant transmissions, or require significant control overhead and intensive computation on nodes. For example, in [15, 17], each node is required to collect and maintain 2-hop neighbor information. Maintaining 2-hop neighbor information for each node incurs extra overhead of the system and the information can be hardly accurate when the mobility of the system is high.

In the paper, we propose an efficient flooding protocol called *vertex forwarding*, which is based on location information of 1-hop neighbor nodes. In our approach, the sender is responsible for nominating the forward nodes of next hop. When a node has a message to flood out, it assumes that it is located at a vertex of virtual hexagonal grid, and the neighbors located at the adjacent vertices will be selected to forward the message. If there is no node located at the vertex, the neighbor nodes nearest to the vertices will be nominated. In such a way the flooding procedure continues hop by hop until every vertex node retransmits the message once. Our scheme is based on this idea to minimize number of forward nodes and collisions.

The rest of this paper is organized as follows. Related work is given in the next section. Section 3 gives the description of *vertex forwarding*. In section 4, we compare performances of different flooding protocols through simulations. Finally, section 5 concludes the work.

2 Related Work

The existing efficient flooding schemes can be classified into three categories based on the information each node keeps: 1) no need of any knowledge of neighbors; 2) 1-hop knowledge of neighbors; 3) 2-hop or more knowledge of neighbors.

Schemes in the first category do not have any assumption on neighborhood knowledge. Authors in [3, 10] showed the serious problem that *pure flooding* causes through analysis and simulations. A probabilistic-based scheme was further proposed to reduce redundant rebroadcasts and differentiate timing of rebroadcasts to avoid collisions. Upon receiving a flooding message for the first time, a node will forward it with probability *P*. The probabilistic scheme includes counter-based, distance-based, location-based and cluster-based flooding schemes. Simulation results showed different levels of improvement over *pure flooding*. This probabilistic scheme was further investigated in [11]. It showed that the success rate curve for probabilistic flooding tends to become linear for the network with low average node degree, and resembles a bell curve for the network with high average node degree. A heuristic method which is based on geometry information was proposed in [29]. Every node schedules a delay time for retransmission according to the distance from it to the strategic location. Closer nodes to the strategic location delay shorter to retransmit the message. In these schemes, a non-redundant transmission might be dropped out. This will cause some nodes in the network failing to receive the flooding message. Besides this deliverability problem, another major concern of these techniques is the difficulties in setting the right threshold value (e.g., retransmission probability, delay time, etc.) in various network situations [12].

Schemes in the second category assume that each node keeps information of 1-hop neighbors. 1-hop neighbor information can be obtained by exchanging the HELLO message periodically. Several different efficient flooding schemes that guarantee 100% deliverability were proposed based on 1-hop knowledge in [6]. There are two strategies for choosing forward nodes: proactive scheme, where each sender nominates a subset of its neighbors to be the next-hop forward nodes, and reactive scheme, where each receiver of a flooding message makes its own decision on whether it should forward the message. The work in [6] also analyzed the performance of the two strategies and concluded the hybrid method achieves a better performance. The flooding with self pruning (*FSP*) scheme proposed in [13] is a receiver-based scheme that uses 1-hop knowledge. In this scheme, a sender forwards a flooding message by attaching all of its 1-hop neighbors to the message. A receiver compares its own 1-hop neighbors with the node list in the message. If all its 1-hop neighbors are already included in the list, it will not forward the message. The work in [14] compared the performance of several flooding schemes. It showed that the improvement of *FSP* is very limited in most of network conditions. A scheme that can achieve the local optimality was proposed in [30]. The union of coverage disks of all 1-hop neighbors plus itself forms an enclosed area, only the neighbor nodes who contribute to the boundary of the area need to forward message. Another notable work of efficient flooding that uses 1-hop neighbor information is *edge forwarding*, proposed in [5]. For each node, its transmission coverage is partitioned into six equal-size sectors. When a node, say *B*, receives a new flooding message from its parent, say *A*, *B* first determines which sector of *A B* is currently in. *B*'s partition lines divide the sector into 6 sub-partitions. If there is at least one node in each sub-partition, *B* does not forward the message.

Most existing flooding schemes that use neighborhood knowledge are based on information of 2-hop or more neighbors. To obtain the information about 2-hop neighbors, one solution is that each node periodically exchanges the list of adjacent nodes with its neighbors. The schemes proposed in [13, 15, 16, 17] are sender-based, while schemes in [7, 18, 19, 20, 21, 22] are receiver-based. In the schemes that use 2-hop neighbor information, a node knows the network topology (connectivity) of 2-hop neighbors. The task for each node to decide the next hop forward nodes is to select the minimal subset of its 1-hop neighbors that can reach all its 2-hop neighbors. A *multi-point relaying* method was proposed in [15, 16], which tries to find the minimal number of forward nodes among the neighbors. Finding the minimal number of forward nodes was proved to be NP-complete [16]. Another important technique is the use of connected dominating set (*CDS*) [7, 23]. A *dominating set* (DS) is a subset of nodes such that every node in the graph is either in the set or is adjacent to a node in the set. A *CDS* is a connected DS. Any routing in MANETs can be done efficiently via *CDS* [7]. Although finding minimal *CDS* (*MCDS*) is NP-hard even in unit disk graph (UDG) [24], some distributed algorithms for computing *MCDS* with approximation ratio have been proposed in [19, 25]. However, maintaining a *CDS* in the network is costly, which is not suitable for flooding operations in highly mobile situations. Generally, the schemes that use 2-hop neighbor information incur high protocol overhead in the network with high mobility and high node density.

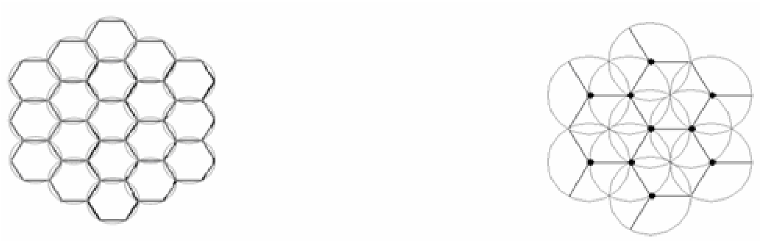
3 Vertex Forwarding Scheme Based on 1-Hop Knowledge

3.1 Basic Idea

The essence of any efficient flooding scheme is to cover all the nodes in the region with a subset of nodes which are connected. Since the coverage range of a node can be represented as a circle, so the efficient flooding can be considered as *the covering problem* which has been well studied. *The covering problem* is stated as follows:

"What is the minimum number of circles required to completely cover a given 2-dimensional space?"

Kershner [28] showed that no arrangement of circles could cover the plane more efficiently than the hexagonal grid arrangement shown in Fig. 1 (a). Initially, the whole space is covered with regular hexagons. The size of hexagons is equal to the transmission range of node, say R, and then circles are drawn to circumscribe them. Obviously, the space can be fully covered by circles in this way. It shows that if each hexagon has a forward node at its center, the network region can be fully covered. However, it does not ensure forward nodes are connected since the distance between adjacent forward nodes is $\sqrt{3}\,R$. One solution to ensure connectivity is to place forward nodes at vertices of hexagons. See the example in Fig. 1(b), the distance between neighbor nodes is R.



(a) Circles located at the center of hexagon (b) Circles located at the center of hexagon

Fig. 1. Covering a 2-dimension space with circles

The basic idea of *vertex forwarding* is to guide the flooding procedure with a virtual hexagonal grid. In our scheme, each node maps itself as a vertex of hexagon grid and keeps the knowledge that who are its adjacent vertex nodes, i.e., the neighbor nodes which are locate at or nearest to adjacent vertices. Before a node floods a message out, it will attach the list of adjacent vertex nodes to the message. The nodes in the list are nominated as forward nodes. Upon receiving a new flooding message, the nodes in the list will select its own adjacent vertex nodes as next-hop forward nodes and retransmit the message. In such a way the flooding procedure continues hop by

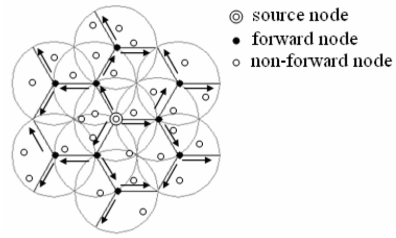


Fig. 2. Example of *vertex forwarding*

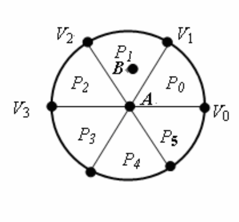


Fig. 3. Transmission coverage partition

hop until every vertex node retransmits the message once. Fig. 2 shows the flooding procedure of *vertex forwarding* in an ideal network situation which has a node at each vertex of hexagonal grid.

3.2 Forward Node Selection

In our scheme, every node keeps the knowledge of its adjacent vertex nodes which forms a forward candidate set. When a node receives a new message, it selects a subset of nodes from its forward candidate set to forward the message. The selection of forward candidate set is based on transmission coverage partition introduced in [5].

Given a node, say *A*, we partition its transmission coverage into six equal-size regions and identify them as P_0, P_1, P_2, P_3, P_4, P_5, respectively. The partition and naming rules are illustrated in Fig. 3. We say a node is *A*'s P_i neighbor, if the node is currently inside partition P_i of *A*, where $0 \le i \le 5$. For instance, in Fig. 3, *B* is *A*'s P_1 neighbor as *B* is located in P_1. Given *A* at location (x_a, y_a) and its 1-hop neighbor *B* at location (x_b, y_b), the Euclidean distance between *A* and *B* is $dist(A, B)=\sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}$. We can determine which partition of *A* contains *B* with some simple computation as follows:

if $x_a \le x_b$ and $y_a \le y_b$, then *B* is in P_0 if $\frac{x_b - x_a}{dist(A, B)} \ge \frac{1}{2}$; otherwise, *B* is in P_1 ;

if $x_a > x_b$ and $y_a < y_b$, then *B* is in P_1 if $\frac{x_a - x_b}{dist(A,B)} \le \frac{1}{2}$; otherwise, *B* is in P_2;

if $x_a > x_b$ and $y_a > y_b$, then *B* is in P_3 if $\frac{x_a - x_b}{dist(A,B)} \ge \frac{1}{2}$; otherwise, *B* is in P_4;

if $x_a \le x_b$ and $y_a \ge y_b$, then *B* is in P_4 if $\frac{x_b - x_a}{dist(A,B)} \ge \frac{1}{2}$; otherwise, *B* is in P_5;

The positions of V_0 … V_5 are the ideal positions of vertex nodes. To formally present our forward candidate set selection algorithm, we have following definitions.

Definition 1. Partition *i* of node *A*, denoted by A_{Pi} , is the set of nodes that are included in partition *i* of node *A*.

Definition 2. Forward Candidate Set of node *A*, denoted by *C*(*A*), is the candidate set of forward nodes that are computed by *A*.

There are at most six candidate nodes in *C*(*A*), $C(A)=\{f_0, f_1, f_2, f_3, f_4, f_5\}$. Ideally, these candidate nodes, $f_0, f_1, f_2, f_3, f_4, f_5$, are on the vertices of *A*'s hexagon. Otherwise, f_i is the nearest node to V_i in the union of two partitions that both border V_i. Take example in Fig. 3 again, since V_0 borders P_0 and P_5, $f_0 = \{u \mid u \in A_{P0} \cup A_{P5}$, and $dist(u, V_0) = \min\{ dist(v, V_0) \mid v \in A_{P0} \cup A_{P5} \}\}$.

Definition 3. Forward Set of node *A*, denoted by *F*(*A*), is the set of nodes that are selected from *C*(*A*) to forward broadcast messages. We have $F(A) \subseteq C(A)$.

Notice that in some cases, vertex node in certain direction may not exist. For instance, if there is no node in both P_0 and P_1, we let $id(f_1)$ as null.

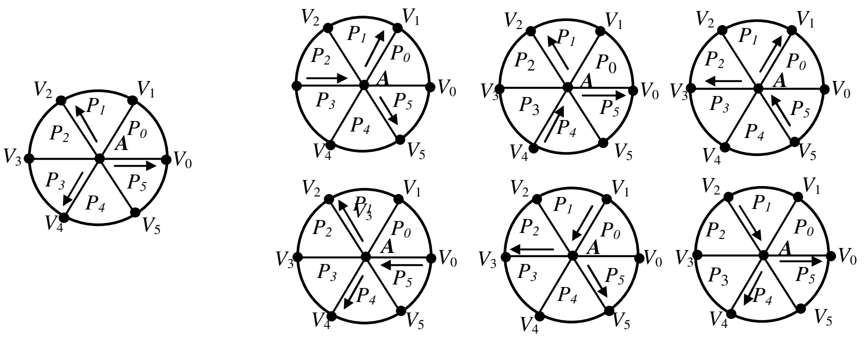
Algorithm 1. Forward Candidate Set Selection Algorithm

```
Output: $C(A) = \{ f_0, f_1, f_2, f_3, f_4, f_5 \}$.
Begin
    for each $V_i$ //$i$=0,…,5.
        if there exist nodes in $A_{Pi} \cup A_{P(i+5)\%6}$
            find the nearest node $f_i$ to $V_i$ in $A_{Pi} \cup A_{P(i+5)\%6}$;
        else
            $f_i$=NULL;
Output $C(A)$.
End
```

Every node in the network computes its forward candidate set by the above algorithm. The remaining problem is how to select forward nodes from this candidate set. We consider the two cases where a node A determines its forward node set $F(A)$:

(1) A is a source node. For simplicity, three vertex nodes in the direction towards V_0, V_2, V_4, are nominated as next hop forward node. That is, $F(A)=\{f_0, f_2, f_4\}$ shown in Fig. 4(a). Node A attaches information of $F(A)$ to the flooding message and broadcasts it out. The information contained in the packet header is not only the IDs of nodes in $F(A)$, but also the directions of them. In this case, $F(A)=\{f_0, f_2, f_4\}$. The information node A attaches is (id (f_0), V_0), (id(f_2), V_2) and (id(f_4), V_4).

(2) A is a forward node. In this case, it will try to select two forward nodes from its forward candidate set. The selection of forward nodes is according to the direction information in the flooding message. The selection rules are defined in Table 1. For example in the first figure of Fig. 4(b), node A receives a flooding message that contains information (id(A) , V_3). According to Table.1, node A nominates f_1, f_5 that are the nearest nodes to V_1, V_5, respectively, as the next hop forward nodes.



(a) Source node broadcasting (b) Forward node broadcasting

Fig. 4. An illustration of forward node Selection

Table 1. Rules of forward node selection

Receiving direction	Forward nodes
V_3	f_1, f_5
V_4	f_2, f_0
V_5	f_3, f_1
V_0	f_4, f_2
V_1	f_5, f_3
V_2	f_0, f_4

Detailed algorithm for computing the forward set is as follows.

Algorithm 2. Forward Set Selection Algorithm

```
Output: $F(A)$.
Begin
    if $A$ is a source node
        $F(A)=\{f_0, f_2, f_4\}$;
    else
        Select two nodes from $C(A)$ into $F(A)$ according to Tab. 1;
    Attach information of $C(A)$ and broadcast $m$ out.
End
```

Based on the forward set selection algorithm, the complete Vertex Forwarding algorithm is as follows.

Algorithm 3. Vertex Forwarding Algorithm

```
Input: a flooding message $m$ from one of its neighbors.
Begin
    if $m$ was received before
        Drop $m$;
    else
        Deliver $m$ to upper layer;
        if $id(A)$ is contained in $m$
            Compute $F(A)$ by running Algorithm 2;
            Attach IDs and directions of nodes in $F(A)$ and broadcast $m$ out.
End
```

4 Simulation

4.1 Simulation Description

To evaluate the performance of our protocol, we run simulation under the *ns-2* with CMU wireless extension. The simulator parameters are listed in Tab. 2. The two-ray ground reflection model is adopted as the radio propagation model. The MAC layer scheme follows the IEEE 802.11 MAC specification. We use the broadcast mode with no RTS/CTS/ACK mechanisms for all message transmissions. The bandwidth of a wireless channel is set to 2M *b*/*s*. CBR(Constant Bit Rate) data traffic is generated.

Notice that some of the schemes require the node to send HELLO message to its 1-hop neighbors periodically. This cost of HELLO message is ignored in our performance study.

Table 2. Simulation parameters

Parameter	Value
Simulator	ns-2
MAC Layer	IEEE 802.11
Data Packet Size	64 bytes
Bandwidth	2 Mb/s
Number of Node	50~1000
Size of Square Area	400,000~4000,000 m^2
Number of Trails	100

Among the existing flooding protocols, we choose following schemes to compare with ours: *Pure flooding*, *edge forwarding* (it requires 1-hop knowledge [5]), and *CDS*-based flooding (it requires 2-hop knowledge). In *CDS*-based scheme [11], a node marks it belong to *CDS* if it has a pair of neighbors which can not directly communicate with each other. After that, a marked node quits CDS if its neighbors are covered by two CDS nodes that are its neighbors with greater IDs. It was proved that the marked nodes form a *CDS* [7]. Notice that all forward nodes that are involved in flooding operation form a CDS in the network. It means that the number of forward nodes that are required in flooding operation is not less than the number of MCDS (Minimum CDS) in the network. So the number of MCDS is the lower bound of the number of relay nodes. Although computing MCDS is NP-hard, there exists a ratio-8 approximation algorithm [25]. This lower bound is further computed to compare with the result of our proposed algorithm. We analyze flooding efficiency in terms of the following two metrics:

Forwarding ratio: Forwarding ratio is the fraction of forward nodes in a flooding operation over the total number of nodes in the network.

Delivery ratio: delivery ratio is the ratio of the nodes that received packets to the number of the nodes in the network for one flooding operation.

4.2 Results and Analysis

4.2.1 Testing the Forwarding Ratio

In this subsection, we study how the forwarding ratio of the flooding techniques is affected by *node density* and *network size* respectively. The results presented in the following figures are the means of 100 separate runs.

Effect of node density: In this study, we use the static network and generate a certain number of nodes from 50 to 500 in step of 50, in each simulation run and place them randomly on a $1000 \times 1000\ m^2$ terrain. The performance data are plotted in Fig. 5. Since in *pure flooding* scheme every node forwards flooding messages, we do not draw its curve. In this figure, we can see that *vertex forwarding* can drastically reduce redundant retransmission of flooding message and significantly outperform edge forwarding and CDS-based schemes. For example, when the number of nodes is 200, the forwarding ratios of *vertex forwarding, edge forwarding* and *CDS*-based schemes are 12.3%, 88.5%, 63.5% respectively. And when the number is 400, the forwarding ratios are 5.9%, 72.4%, 67.3% respectively. Our scheme is distinctly efficient because that in the coverage of a node, we at most select 3 nodes to forward the message regardless if they can cover all 2-hop neighbors or not. In contrast, in the other two schemes, forward node set must cover all the 2-hop neighbors, so they are prone to choose more 1-hop neighbors as forward nodes.

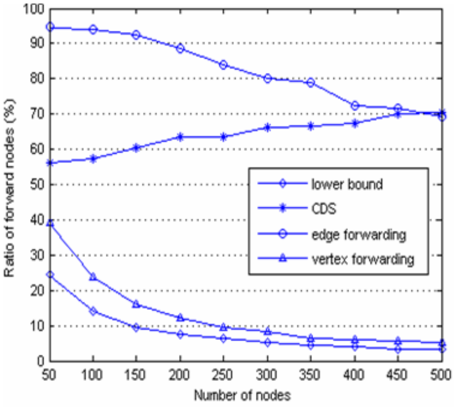


Fig. 5. Ratio of forward nodes VS. number of nodes

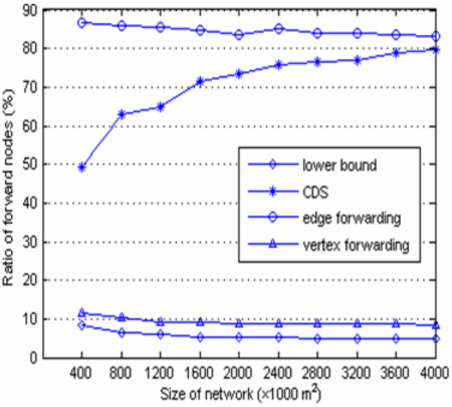


Fig. 6. Ratio of forward nodes VS. network size

Effect of network size: In this simulation, we increase the area of the network region, from 400,000 to 4,000,000 m^2 while the node density is fixed at 4000 $m^2/node$. We fix the radio transmission radius at 250m. The simulation results are plotted in Fig. 6. We observe that our flooding scheme and Edge Forwarding scheme are both highly scalable with respect to the network size. In contrast, performance of CDS-based scheme is better in a smaller network but becomes worse when network size increases. Thus, it is not a scalable flooding scheme. Once again, our flooding scheme outperforms other schemes, and outputs the closest results to the lower bound in the comparison.

4.2.2 Testing the Delivery Ratio

In this subsection, we study how the delivery ratio of the proposed techniques is affected by *traffic load* and *node density* respectively. The results presented in the following figures are the means of 100 separate runs.

Effect of node density: In this simulation, a certain number of nodes, from 50 to 500, are randomly placed on a 1000× 1000 m^2 area. The network load is set to 10Pkt/s and lasts for 100 seconds. The performance data are plotted in Fig. 7. We can see that the delivery ratio of vertex forwarding is almost 100% and not affected by node density varied. In contrast, the performance of both pure flooding and edge forwarding degrades as node density increase.

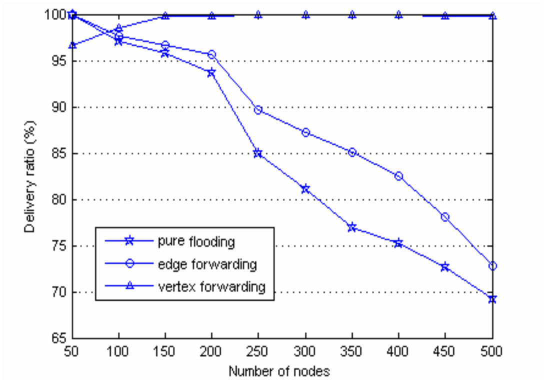


Fig. 7. Broadcast delivery ratio VS. number of nodes

Effect of traffic load: In this simulation,. 200 nodes are randomly placed on a 1000×1000 m^2 area and the transmission range is fixed at 250 *m*. We change the network traffic load from 1Pkt/s to 25Pkt/s and each simulation is run for 100 seconds. The delivery ratio in Fig. 8 shows a few remarkable results. Firstly, when traffic load is light (<10Pkt/s), the delivery ratio of each flooding schemes is equal or close to 100%. Secondly, the performance of both *pure flooding* and *edge forwarding* degrades quickly as traffic ratio increase. This is because that when traffic load is heavy, different flooding messages may concurrence in the network, which may incurs more collisions. Although delivery ratio of vertex forwarding also degrades when traffic load is getting heavier, it is not as sensitive to traffic load as the other two schemes.

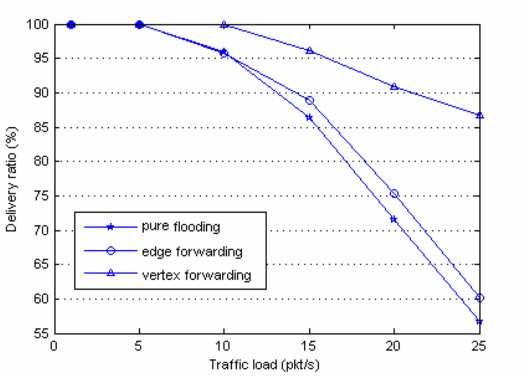


Fig. 8. Broadcast delivery ratio VS. traffic load

5 Conclusions

We have presented a location aided flooding protocol which is based on 1-hop knowledge. We call the technique as *vertex forwarding*, alluding to the fact that it nominates nodes to forward message unless they are located at or closest to the vertices of the broadcast coverage. It provides an adequate solution to reduce redundant retransmissions in the wireless ad hoc network. Simulation results show that our scheme uses less forward nodes, obtain high delivery ratio and is highly scalable, compared with the existing schemes.

References

1. Ho, C., Obraczka, K., Tsudik, G., Viswanath, K.: Flooding for Reliable Multicast in Multihop Ad Hoc Networks. In: Proc. of the Int'l Workshop on Discrete Algorithms and Methods for Mobile Computing and Communication, pp. 64–71 (1999)
2. Jetcheva, J., Hu, Y., Maltz, D., Johnson, D.: A Simple Protocol for Multicast and Broadcast in Mobile Ad Hoc Networks. In: Internet Draft: draft-ietf-manet-simple-mbcast-01.txt (2001)
3. Ni, S., Tseng, Y., Chen, Y., Sheu, J.: The broadcast storm problem in a mobile ad hoc network. In: Proc. of ACM/IEEE MOBICOM 1999, pp. 151–162 (1999)
4. Sinha, P., Sivakumar, R., Bharghavan, V.: Enhancing ad hoc routing with dynamic virtual infrastructures. In: IEEE INFOCOM 2001, pp. 1763–1772 (2001)
5. Cai, Y., Hua, K.A., Phillips, A.: Leveraging 1-hop Neighborhood Knowledge for Efficient Flooding in Wireless Ad Hoc Networks. In: IPCCC. 24th IEEE International Performance Computing and Communications Conference, Phoenix, Arizona (2005)
6. Yang, C.-C., Chen, C.-Y.: A Reachability-Guaranteed Approach for Reducing the Broadcast Storms in MANETs. In: Proceedings of IEEE Semiannual Vehicular Technology Conference (2002)
7. Wu, J., Li, H.: On Calculating Connected Dominating Set for Efficient Routing in Ad Hoc Wireless Networks. In: DiaLM. Proc. of the 3rd Int'l Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, pp. 7–14 (1999)
8. Wieselthier, J.E., Nguyen, G.D., Ephremides, A.: On the Construction of Energy-Efficient Broadcast and Multicast Trees in Wireless Networks. In: IEEE INFOCOMM'2000 (2000)
9. Li, D., Jia, X., Liu, H.: Energy efficient broadcast routing in ad hoc wireless networks. IEEE Trans on Mobile Computing 3(2), 144–151 (2004)
10. Tseng, Y., Ni, S., Shih, E.Y.: Adaptive Approaches to Relieving Broadcast Storms in a Wireless Multihop Mobile Ad Hoc Networks. In: Proc. of ICDCS 2001, pp. 481–488 (2001)
11. Sasson, Y., Cavin, D., Schiper, A.: Probabilistic broadcast for flooding in wireless mobile ad hoc networks. In: Swiss Federal Institute of Technology, Technical Report IC/2002/54 (2002)
12. Sun, M.T., Feng, W.C., Lai, T.H.: Location Aided broadcast in wireless ad hoc networks. In: Proc. of GLOBECOM 2001 (2001)
13. Lim, H., Kim, C.: Multicast Tree Construction and Flooding in Wireless Ad Hoc Networks. In: MSWIM. Proc. of the ACM Int'l Workshop on Modeling, Analysis and Simulation of Wireless and Mobile System, pp. 61–68 (2000)
14. Williams, B., Camp, T.: Comparison of Broadcasting Techniques for Mobile Ad Hoc Networks. In: Proc. of MOBIHOC 2002, pp. 205–214 (2002)
15. Qayyum, L.A., Viennot, L.: Multipoint relaying: An efficient technique for flooding in mobile wireless networks. In: HICSS'2001. 35th Annual Hawaii International Conference on System Sciences, IEEE Computer Society, Los Alamitos (2001)

16. Qayyum, L.V., Laouiti, A.: Multipoint Relaying for Flooding Broadcast Messages in Mobile Wireless Networks. In: Proceeding of the 35th Hawaii International Conference on System Sciences (2002)
17. Lou, W., Wu, J.: Double-Covered Broadcast (DCB): A Simple Reliable Broadcast Algorithm in MANETs. In: Proc. of INFOCOM 2004 (2004)
18. Chen, K.J., Balakrishnan, H., Morris, R.: Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks. In: Proc. of MOBICOM 2001, pp. 85–96 (2001)
19. Peng, W., Lu, X.: On the Reduction of Broadcast Redundancy in Mobile Ad Hoc Networks. In: Proc. of MOBIHOC 2000 (2000)
20. Stojmenovic, M.S., Zunic, J.: Dominating Sets and Neighbor Elimination Based Broadcasting Algorithms in Wireless Networks. IEEE Transactions on Parallel and Distributed Systems 13(1), 14–25 (2002)
21. Sucec, J., Marsic, I.: An Efficient Distributed Network-Wide Broadcast Algorithm for Mobile Ad Hoc Networks. In: Rutgers University, CAIP Technical Report 248 (2000)
22. Wu, J., Dai, F.: Broadcasting in Ad Hoc Networks Based on Self-Pruning. In: Proc. of INFOCOM 2003 (2003)
23. Dai, F., Wu, J.: An Extended Localized Algorithm for Connected Dominating Set Formation in Ad Hoc Wireless Networks. IEEE Trans. Parallel Distrib. Syst. 15(10), 908–920 (2004)
24. Marathe, M.V., Breu, H., Hunt III, H.B., Ravi, S.S., Rosenkrantz, D.J.: Simple heuristics for unit disk graphs. Networks 25, 59–68 (1995)
25. Wan, P.-J., Alzoubi, K., Frieder, O.: Distributed Construction of Connected Dominating Set in Wireless Ad Hoc Networks. In: Proc. IEEE INFOCOM, vol. 3, pp. 1597–1604 (2002)
26. Kowalski, D.R., Pelc, A.: Deterministic Broadcasting Time in Radio Networks of Unknown Topoloby. In: Proc. of the 43rd Annual IEEE Symposium on Foundations of Computer Science (2002)
27. Gandhi, R., Parthasarathy, S., Mishra, A.: Minimizing Broadcast Latency and Redundancy in Ad Hoc Networks. In: MOBIHOC 2003. Proc. of the Fourth ACM Int. Symposium on Mobile Ad Hoc Networking and Computing, pp. 222–232 (2003)
28. Kershner, R.: The number of circles covering a set. Amer. J.Math 61 (1939)
29. Paruchuri, V.K, Durresi, A., Jain, R.: Optimized Flooding Protocol for Ad hoc Networks. In: proc. of IEEE WCNC 2003 (2003)
30. Liu, H., Jia, X., Wan, P., Liu, X., Yao, F.: A Distributed and Efficient Flooding Scheme Using 1-hop Information in Mobile Ad Hoc Networks. IEEE Transactions on Parallel and Distributed Systems 18(5) (2007)

A MAC-Layer Retransmission Algorithm Designed for Zigbee Protocol

Yi Li[1], Dongliang Xie[1], Jian Ma[2], and CanFeng Chen[2]

[1] State Key Lab of Networking and Switching Technology
Beijing University of Posts & Communications, Beijing 100876, China
xiangtanliyi@sohu.com, xiedl@bupt.edu.cn,
{jian.J.Ma,CanFeng.Chen}@Nokia.com
[2] Nokia Research Center, Beijing
Nokia House 1, No.11, He Ping Li Dong Jie, Beijing 100013, P.R.China.
jian.j.ma@nokia.com

Abstract. A typical scenarios was considered , in which an event, such as a mobile sink arriving to collect data via the wireless sensor network, initiates the collection of one packet of data from each node in the surrounding cluster. The node requesting the beacon synchronizes its time clock and purpose to the cluster and assumes the role of clusterhead. All nodes covers an area with a radius of several hundred meters, to within a few microseconds. We exploit this ability to extend the Contension Windows to improve retransmission algorithms in Zigbee MAC protocols. The result is a retransmission algorithm that uses a Extended Shared Contention Window (ESCW) that is easy to implement and results in fewer collisions than retransmission algorithms that use Binary-Exponential-Backoff (BEB). We show via numerical and simulation results that an ESCW-based Zigbee protocol performs significantly better – in terms of energy usage, throughput, and the time to complete the data collection task – than standard Zigbee (IEEE802.15.4).

1 Introduction

A common scenario in sensor networks is the collection of information from a cluster of wireless sensor nodes by a clusterhead (CH) that is either a mobile sink that is passing by or a regular node that has been assigned the role of CH. The CH's task is to quickly gather one packet of data from each sensor node within range.

Two of the most critical performance measures for scenarios like those above are the time required to gather one packet of data from each sensor node and the energy the sensor nodes expend before successfully transmitting their data. Using these two measures, we determine the performance of several approaches to the design of the MAC layer for this scenario.

We demonstrate how the performance of ZigBee can be significantly improved in this situation by redesign of their retransmission control algorithms. The new algorithm replaces the standard algorithm used in current MAC algorithms with a Extended, Shared Contention Window (ESCW) retransmission algorithm. In the

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 314–325, 2007.

ESCW approach, there is a time window, called the contention window, that is initiated by the CH's beacon. All nodes with packets schedule their transmission or retransmission attempts within this contention window. Packets involved in collisions in a given contention window are rescheduled at random times in the next contention window. This process repeats, one window at a time, until all packets have been collected. This new approach to managing retransmissions in sensor networks with random access based MAC layers is simpler than others and offers better performance. Simulation results show that this new protocol outperforms standard ZigBee (802.15.4). It results in fewer collisions and thus uses less energy and wastes less time on the channel.

Paper organization: Related work is reviewed in Section 2. In Section 3, we investigate the performance of Zigbee in a clustered one-hop environment; propose a new ESCW-based protocol. The numerical and simulation results in Section 4 show that this ESCW-based protocol performs significantly better than standard Zigbee. Section 5 is the conclusion of the paper.

2 Related Work

Clustered architectures for sensor networks have been studied extensively [1-7]. Results on the design of Medium Access Control (MAC) protocols for these architectures are available, with the most prominent example being an option in the proposed ZigBee standard, 802.15.4. In this option, a synchronization beam from a PAN-Coordinator (CH) initiates a superframe on the wireless channel. Nodes randomly schedule their first transmissions in the slots in this superframe. Subsequent new transmissions and retransmissions follow slotted-CSMA/CA procedures.Failed transmissions in ZigBee can be rescheduled for either the current or subsequent superframes. In ESCW-based protocols, failed transmission attempts in the current contention window must wait until the next contention window to attempt a retransmission.

In [14], a MAC protocol using a scheduling algorithm for data collection is proposed. The CH uses information about interference patterns created by the physical location of each non-CH node within the cluster to schedule simultaneous, collision-free transmissions. While [14] determined a lower-bound on the performance of other MAC protocols, its implementation depends on knowledge of the physical location of each node. This may not be possible in many cases and relies on either polling by the CH or well-synchronized clocks in the sensor nodes. In [3], S-MAC, a standard 802.11-based MAC protocol designed for wireless sensor networks, has been proposed. Sensor nodes running S-MAC form virtual clusters. Nodes within a cluster use shared schedules to generate traffic-adaptive sleep/wake up cycles, thus reducing the energy used for overhearing and idle listening. In contrast, our work focuses on an event driven scenario in which all sensor nodes are first activated by an event and then transmit data to the CH. These results could possibly be combined with S-MAC to achieve energy savings if the resulting increase in delay is not of concern. The MAC standard has been considered for sensor nets because of its low cost and wide availability. It does not require synchronization, as in [14] and ZigBee [12], but the cost is wasted energy due to collisions and delays due to idle channel time. More generally, standard Zigbee was designed for homogeneous wireless peer-to-peer communication amongst large numbers of nodes, so it is far from optimal for the clustered structure that is often present in sensor networks.

The retransmission algorithm that is most closely related to the ESCW-based algorithm developed in this paper is the MACAW protocol [15]. MACAW assumes: (i) that each user in the network can receive every other user's transmission; and (ii) every packets' header contains the value of the backoff window that was used when that packet was transmitted. All users set their backoff window counters to this value when they receive a successful packet, and thus use a common backoff window value after each successful transmission. A similar capability is achieved under ESCW retransmission algorithms by having the CH transmit the length of the contention window to be used by all nodes at the same time that it transmits the synchronization beam , and, unlike MACAW, we only require that all nodes be able to receive packets correctly from and transmit successfully to the CH. Furthermore, in ESCW algorithms the same size contention window is used by all nodes until it is changed by the CH, thus ensuring fairness at all times, not just after a successful transmission.

Recent work on retransmission algorithms includes the Fast Collision Resolution (FCR) and Fairly Scheduled FCR (FS-FCR) algorithms proposed in [16]. In these algorithms, feedback from the channel on the number of idle slots, collisions, and successful transmissions is used by individual nodes to adjust a "multiplicative-increase, linear-decrease" backoff algorithm. At any point in time, different users may thus be using different size backoff windows. A modified collision resolution algorithm called Gentle DCF (GDCF) is proposed to enhance the performance of IEEE standard 802.11 DCF [6]. In this case, the backoff window size is halved after c consecutive successful transmissions and an optimal value for c is proposed. Other recent work on retransmission strategies [7] considers an event-driven system in which N nodes that have detected an event, such as a fire, are all trying to send an alert packet to a base station. In our case, we fix the number of nodes, and assume all transmission attempts are uniformly distributed over a contention window of length W.

3 ESCW-Based Performance in a Cluster

In the scenario assumed in this paper, nodes become active after the mobile sink or another node acting as the CH announces the beginning of the data collection process. The traffic pattern that occurs after this event is not a stochastic arrival process; instead, each of the N nodes in the one-hop sensor cluster has data to transmit and they initiate the channel access procedure at roughly the same time. We also assume that each sensor will transmit only one packet to the CH and then remain silent until the arrival of the next request. Our goal is to find a MAC protocol that is well-suited to these situations in clustered mobile wireless sensor networks.

In the class of wireless sensor network protocols, Zigbee is currently the dominant protocol in terms of presence in the commercial market. Thus, in the following subsections, we determine the effect that the use of an ESCW-based retransmission algorithm has on Zigbee .We assume that the beacon-enabled mechanism in Zigbee is used. All the settings in Zigbee not affected by the retransmission control algorithm are kept the same for the two approaches in order to determine the improvement due to the change in retransmission strategy. These parameters are used in all numerical work and all NS-2 simulations in this paper.

To determine the performance of ESCW-based Zigbee, and to compare it against standard Zigbee, we define the following variables:

W : The size of the contention window; measured in slots. In ESCW-based Zigbee, each node maintains its own contention window. For each transmission attempt, the node randomly chooses a slot in a window of size .In ESCW-based Zigbee, all users randomly schedule their transmission attempts in the shared window of size W slots. Fialied transmission attempts are delayed and then reattempted in the next window of size W .

f_{NW} :The cost function when a contention window of W slots is used and there are N nodes.

P_S : The probability that all nodes can be successfully transmitted without any collisions.

P_{Wnrc} The probability that n nodes transmit successfully and the other r nodes are involved in c collisions and the contention window size is W.

T_C : The duration of an RTS collision; defined as . $T_C = T_{RTS} + T_{EIFS} + T_{DIFS}$

T_D : The duration of a successful data packet transmission; defined as: $T_D = T_{RTS} + T_{CTS} + T_{ACK} + 3 * T_{SIFS} + T_{DIFS}$

$T_E(N,W)$: The time required to collect one packet from each node in the one-hop cluster when there are N nodes and the size of the CW is W .

$T_W(N,W)$: The total time wasted on collisions and idle slots while emptying a cluster given N and W ; thus, $T_W(N,W) = T_E(N,W) - N * T_D$

T_i : The duration of the i -th contention window during the collection of one packet from each node.

T_{EIFS} : The duration of an Extended InterFrame Space; defined as $T_{EIFS} = T_{ACK} + T_{SIFS} + T_{DIFS}$

3.1 A Cluster Using Standard Zigbee Retransmissios: Cross-Stage Collisions

If the nodes in the cluster are all using Standard Zigbee, we assume that they all initiate their transmission process on the cue of the beacon from the CH. Every node thus senses the channel idle for one InterFrame Space period and transmits its packet. The common destination node, the CH, will then see the collision of all of these packets. This collision results in an Extended InterFrame Space. At this point, each node will randomly choose a slot for another attempt in a contention window of length $2W$ slots. This is the same window used by all nodes since they have all experienced one unsuccessful transmission attempt.

Define $\overline{W}=\left(W_{1,1},W_{1,2}\cdots\cdots W_{1,N}\right)$ to be the vector of backoff slots the N users have chosen; where each element of the vector lies in $\left[0,W-1\right]$Without loss of generality, we assume that $W_{1,1}\leq W_{1,2}\leq\cdots\leq W_{1,N}$ Depending on the relationships among the W_{1,n_s} there are two cases:

Case 1: *No collisions in the contention window.* If $N<W$and the W_{1n} s are all distinct, then it is possible for all data packets to be successfully transmitted. This occurs with probability

$$P_s=\frac{W(W-1)\cdots(W-N+1)}{W^N}=\frac{(W)_N}{W_N}$$

where, by convention, $(W)_N=W(W-1)\cdots(W-N+1)$ In this case of no collisions, the total time to empty the cluster will be $T_E=T_C+N.T_D+W_{1,N}$. The mean value of T_E in this case can easily be shown to be: $E(T_E \mid no\ collisions)=T_C+N.T_D+E(W_{1,N} \mid no\ collision)$ where, recalling that there is a slot numbered zero:

$$E\left(T_E \mid no\,collision\right)=\sum_{x=N-1}^{W-1}x.P\left(W_{1,N}=x \mid no\,collision\right)=\sum_{x=N-1}^{W-1}x.\frac{N.\binom{x}{N-1}.(N-1)!}{W^N}$$

Case 2: *Collisions in the contention window.* With probability, $1-P_S$ there will be at least one collision. In this case, at least two of the nodes' $W_{1,N}s$ assumed the same value. When their backoff slot counters reach zero, simultaneous transmissions from all nodes with this , $W_{1,N}$ will lead to a collision, which causes all busy nodes to freeze their backoff counters for one EIFS period. The nodes that collided will immediately choose new slots to reattempt their transmissions. They choose slots at random from a window $\left[0,2W\right]$ For simplicity, we assume that one collision occurs, that two nodes were involved, and that these nodes' first-stage backoff slots are , $W_{1,n}=W_{1,n+1}=x$, $0\leq x\leq W-1$, $1\leq n\leq N-1$,We assume their second-stage backoff counters are $W_{2,n}$ and $W_{2,n+1}$, with both in the interval $\left[0,2W-1\right]$. Since $W_{2,n}$ and $W_{2,n+1}$ are uniformly distributed over $\left[0,2W-1\right]$ these two nodes schedule their transmission attempts in the time interval$\left[x,x+2W-1\right]$ This time interval overlaps by $W-x$ slots with the first-stage backoffs of the rest of the busy nodes. It is thus possible for these packets that are in their second backoff stage to collide with those that are still in their first backoff stage. We call such an event a "*cross-stage*" collision.

By the definition of the EIFS, cross-stage and other collisions lead to long idle times on the channel, decreasing the channel throughput significantly. While increasing the window size can alleviate the problem, cross-stage collisions can not be completely prevented in standard Zigbee. It is also difficult to modify BEB retransmission algorithms to minimize these cross-stage collisions, maximize the throughout, or minimize the per-packet unfairness of the scheme.

3.2 A Cluster Using Zigbee with ESCW-Based Retransmissions

In this section we give the prediction algorithm for one mobile sink and the location update methods as well. We assume a strongly connected network with one mobile sink. Some of the assumptions that we have made are:

3.2.1 The New Retransmission Algorithm

Due to the clustered architecture of the sensor network being considered, synchronization of all nodes is possible and is assumed throughout the remainder of this paper. The synchronization can be achieved via broadcasts from the CH when it announces its intention to collect packets from the cluster and by periodic synchronization updates from the CH.

With knowledge of the number of sensor nodes in the cluster, the CH will broadcast a control packet that specifies a contention window of size W to be used by all nodes in the cluster. We assume that all nodes in the cluster correctly receive the packet from the CH with synchronization and contention window size information. The transmission of data packets from every sensor node to the CH begins at time $t = 0$ with each node selecting a random number n W that is uniformly distributed over $[0, W - 1]$. We assume each node will maintain two counters. One is a backoff counter with starting value W_n; the other counter starts with value W and is called the stage counter. All nodes will start sensing the channel at time , $t = 0$ decreasing both their backoff and stage counters, after each slot period and freezing their counters when a transmission is detected on the channel.

When a node's backoff counter reaches zero, the node will transmit its RTS packet to the CH.If no other node has chosen the same slot in the current contention window, the node will successfully transmit its packet following the procedure of standard Zigbee. The remaining nodes will activate their counters after one DIFS period after the packet transmission ends.Collisions will occur if two or more nodes choose the same random backoff slot. Contrary to the standard Zigbee, though, the nodes involved in the collision will not generate a new random backoff slot until their stage counters reach zero. This ensures that all scheduled attempts to transmit RTS packets take place before any are rescheduled because of collisions.

For all the nodes within the cluster, since each node can detect any transmissions to the CH, a collision will result in an EIFS idle time on the channel while the counters of every node are frozen. With synchronization of all the nodes in the cluster, the stage counters in every node will reach zero simultaneously. At this point, the nodes that suffered collisions will generate a new random backoff slot that is uniformly distributed over the next contention window of length W and repeat the process just described in this new window. The goal of this retransmission algorithm is to use synchronization to enable all nodes involved in collisions in each contention window to generate their next random backoff slot simultaneously at the beginning of the next contention window, thus preventing the “cross-stage” collisions that can occur on standard Zigbee.

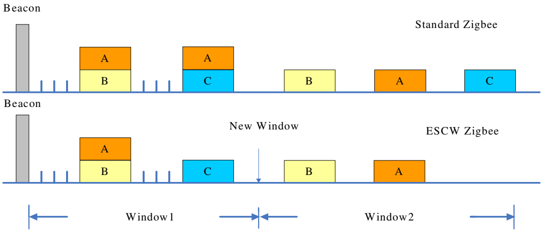


Fig. 1. Channel activity in standard Zigbee vs. ESCW-based Zigbee

We illustrate the difference between the standard and ESCW-based Zigbee in Fig. 1. Assume there are three nodes: A, B and C . Nodes A and B collide during their first backoff stage. With standard Zigbee, A and B then generate their new backoff slots immediately and simultaneously. A picks a small backoff slot that causes it to collide with node C so they both need to generate new new backoff slots. With ESCW-based Zigbee, A and B are prevented from generating new backoff slots at the time they collide; instead, they wait to generate new backoffs until the end of the current contention window. By then, node C has successfully transmitted its packet. By following a ESCW-based retransmission strategy and waiting until the next contention window to reschedule their packets, nodes A and B avoided colliding with node C .

We next analyze the performance of this protocol in the case of a single, one-hop cluster. Theanalysis can be extended to the multiple-hop cluster case if each ring is emptied in turn, with simultaneous emptying of rings and ring sectors that do not interfere with each other [14]. We assume that the total number of non-CH nodes is N and the length of the contention window is W

3.2.2 Performance of ESCW-Based Zigbee

Define $\overline{W}=\left(W_{1,1},W_{1,2}\cdots\cdots W_{1,N}\right)$ to be the sorted random backoff slot vector, where each W is uniformly distributed over $\left[0,W-1\right]$. We determine the time to empty the whole cluster,called T_E This is the time required for the CH to collect one data packet from every non-CH node within the cluster. Before providing a general form for the density function of , T_E we analyze two special cases:

Case 1: No collisions. In this case, all packets are successfully transmitted in the first contention window. This occurs with probability .

$$P_s=\frac{W(W-1)\cdots(W-N+1)}{W^N}=\frac{(W)_N}{W_N}$$

The total time T_E in this case is given by $T_E = W + N.T_D$. , where T_D D T is the same as in standard Zigbee.

Case 2: Collisions occur in the first contention window but no collisions in the second contention window.

Assume that one or more collisions occur in the first contention window, and that all nodes that collided successfully transmit their packets in the second contention window. Define the successful nodes vector $\overline{N} = (N_1, N_2)$ where N_1 and N_2 are random variables that stand for the number of packets successfully transmitted in the first and second contention windows, respectively. Thus, . $N_1 + N_2 = N$. Assume that C_1 , the number of collisions in the first contention window, results in the 2 N nodes that must retransmit. Then the random variable C_1 takes values in the set $\left\{1, 2, 3 \cdots \left\lfloor \frac{N_2}{2} \right\rfloor\right\}$. In this case, the total time to empty the cluster is:

$$T_E = T_1 + T_2 = W + N_1.T_D + C_{1.}T_C + W + N_2.T_D = 2W + N_1.T_D + C_1.T_C \quad (1)$$

$$P\{N = n_1, C_1 = c_1, no\ collision\ in\ 2nd\ round\} = \frac{\binom{N}{n_1}.(W)_{n_1}.(c_1!).s_2(N - n_1, c_1)}{W^N}.\frac{(W)_{n_2}}{W^N}$$

We now analyze the general case. Assume that a total of contention windows are required to empty the cluster, and assume that the vector of the number of successfully transmitting users in each contention window is $\overline{N} = (N_1, N_2, \ldots N_I)$, where . $\sum_{i=1}^{I} N_i = N$ .For simplicity, we also define an associated random vector , $\overline{R} = (R_{1,} R_2, \cdots R_{I-1})$ where , $R_i = \sum_{j=i+1}^{I} N_j$, $i = 1, 2, \ldots I - 1$ .Each R_i is the number of nodes after the i -th contention window that still have to transmit their packets. We also define the collision vector $\overline{C} = (C_{1,} C_2, \cdots C_{I-1})$ where C_i , the number of slots in the i -th contention window in which collisions occur, can assume the values , $1, 2, \ldots \left\lfloor \frac{N_i}{2} \right\rfloor$, $i = 1, 2, \ldots I - 1$. So $\overline{N}$, $\overline{C}$ and $\overline{R}$ occurs with probability:

$$P\{(N_1 = n_{1,} N_2 = n_2, \ldots, N_I = n_I), (C_1 = c_1, C_2 = c_2, \ldots C_{I-1} = c_{I-1}), (R_1 = r_1, R_2 = r_2, \ldots, R_I = r_I)\}$$

$$= \frac{\binom{N}{n_1}.(W)_{n_1}.\binom{W-n_1}{c_1}.(c_{1!}).S_2.(r_{1,}c_{1})}{W^N} \cdot \cdots \cdot \frac{\binom{n_1+r_1}{n_1}.(W)_{n_1}.\binom{W-n_i}{c_i}.(c_{i!}).S_2.(r_{i,}c_i)}{W^{n_1+r_1}} \cdot \cdots \cdot \frac{(W)_{n_I}}{(W)^{n^I}}. \quad (3)$$

This outcome will lead to a total time to empty the cluster of:

$$T_E=\left(N,W\right)=\sum_{i=1}^{I}T_i\sum_{i=}^{I}(W+c_i.T_C+n_i.T_D)+(W+n_I.T_D)=I.W+\left(\sum_{i=1}^{I-1}C_i\right)T_C+N.T_D \tag{4}$$

In (4), $T_E\left(N,W\right)$ is a constant once I , $\overline{n}$ and $\overline{c}$ have been specified. The number of contention windows required to empty the cluster is a random variable whose value is uniquely determined by $\overline{n}$ and $\overline{c}$. Together with (3), we know that:

$$P\left\{T_E=\left(N,W\right)=I.W+(\sum_{i=1}^{I-1}C_i)T_c+N.T_D,\overline{N}=\overline{n},\overline{C}=\overline{c}\right\}$$

$$=\frac{\binom{N}{n_1}.(W)_{n_1}.\binom{W-n_1}{c_1}.(c_{1!}).S_2.\left(r_{1,}c_{_1}\right)}{W^N}.\cdots.\frac{\binom{n_1+r_1}{n_1}.(W)_{n_1}.\binom{W-n_i}{c_i}.(c_{i!}).S_2.\left(r_{i,}c_i\right)}{W^{n_1+r_1}}.\cdots\frac{(W)_{n_I}}{(W)^{n^I}}$$

Given the total number of nodes, , N and a contention window of size , W the mean time wasted before the cluster is empty, $E\{T_E(N,W)\}$, can also be used to evaluate the performance of the algorithm. Given the number of successful nodes in the first contention window, $E\{T_E(N,W)\}$can be found via the following recursion:

$$E\{T_E(N,W)\}=\sum_{n_1=0,n_1\neq N-1}^{N}\sum_{C_1=1}^{\left\lfloor\frac{N-n_1}{2}\right\rfloor}(W+c_{1.}T_C+E\{T_E(N-n_1,W)\}).P_{W,N-n_1},c_1 \tag{5}$$

4 Performance Evaluation and Simulation

We now present numerical and simulation results for the performance of the ESCW-based Zigbee protocols. In all cases, the time scale is normalized by a slot time, which is equal to 10μs. Note that the time devoted to successful transmissions on the channel is common to all protocols; namely, $N.T_D$ Minimizing the wasted time T_W will thus minimizedelay and maximize channel utilization.

Fig. 2 shows that overall channel throughput can be as high as 0.92 for all four cases in the figure, thus demonstrating the high channel utilization that can be achieved with an ESCW-based protocol. We further note that the throughput is not very sensitive to W the size of the contention window, especially when the number of nodes is large. For example, when , $N=200$ choosing $W=2^9$ slots which is almost same as the other contention window size, provides almost the same channel throughput. This shows that the performance of the protocol is not overly sensitive to the choice of W.

The wasted time while empting the cluster consists of two terms: the time due to collisions and the time due to idle slots. While increasing the size of the contention window may reduce the probability of collisions in each slot, it also increases the time until the end of the contention window. On the other hand, decreasing the size of the contention window will decrease the time spent waiting to the end of the window while increasing the time wasted due to collisions. The tradeoff between these two effects leads to an optimal contention window size for each fixed number of sensor nodes within the cluster, as shown in Fig. 3.

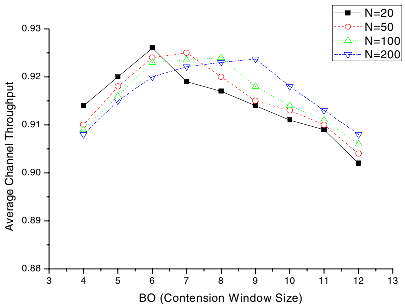


Fig. 2. Simulations for the ESCW-based Zigbee protocol that show the average channel throughput while emptying the cluster for different sizes of the contention window. N is the number of nodes in the cluster.

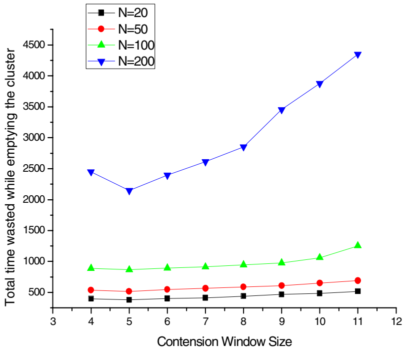


Fig. 3. Simulations for the ESCW-based Zigbee protocol that show the average wasted time,while emptying the cluster for different sizes of the contention window. N is the number of nodes in the cluster.

From fig4, it is clear that the ESCW-based Zigbee protocol utilizes the channel better than stanrdard ZigBee. Its advantage increases as the number of nodes increases. The reason for this improved performance is the elimination in the ESCW approach of "crossstage" collisions, a phenomenon that becomes dominant in ZigBee as the number of nodes increases. Though ESCW-based ZigBee may prevent "crossstage" collisions, some portion of the superframe must be dedicated to the Contention Access Period (CAP); furthermore, each node must reserve an integer number of time slots for data packet transmission. These requirements create significant overhead, which shows up in the figure as inefficient use of the channel.

Energy-efficiency is also a critical issue in sensor networks. Note that the performance improvement of ESCW-based Zigbee over standard ZigBee comes from the elimination of "cross-stage" collisions. Each collision involves at least two transmissions, and nodes consume energy even while waiting until they can retransmit, so we can conclude that ESCW-based ZigBee is more energy-efficient than standard ZigBee. We apply these ratios and calculate the total energy consumed to empty a cluster with different protocols. The results are shown in Fig. 5.

In Fig. 5, the energy consumed is normalized by the energy consumed by a node when it is idle for 1000 microseconds. From Fig. 5, we can see that ESCW-based Zigbee is the most efficient.

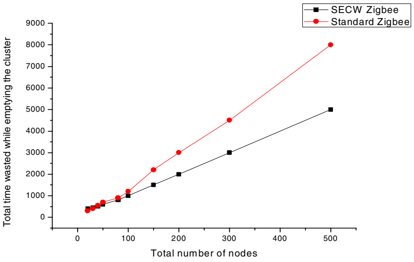


Fig. 4. Simulations comparing total time while emptying the cluster for the ESCW-based Zigbee, standard Zigbee

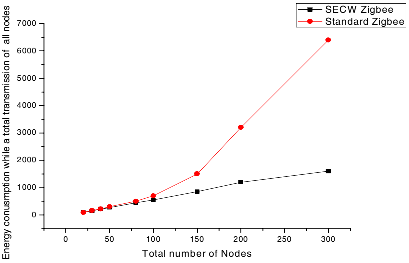


Fig. 5. Simulations comparing total energy consumption to empty the cluster for the ESCW-based Zigbee, standard Zigbee

5 Conclusion

In this paper, we proposed a ESCW retransmission algorithm for random access MAC protocols for clustered sensor networks. We studied the performance of this new algorithm in Zigbee via both probabilistic analysis and simulation. We showed via analysis and simulation that an ESCW-based Zigbee protocol performs significantly better – in terms of energy, wasted time of collecting data from cluster'node and throughput – than standard Zigbee protocol. Our future work in this area will consider the best approach to extending this new backoff scheme to the multi-hop case, which will require synchronization of nodes in outer rings. This can be achieved by higher power broadcasts from the CH or by multi-hop synchronization algorithms that have been proposed for sensor networks.

Acknowledgements. This research was supported partly by Nokia-BUPT MVAS Joint Fund, national "863"plan(No. 2007AA01Z222, 2006AA01Z229), Programme Sino-Français de Recherches Avancées (PRA-SI07-01).

References

1. Bandyopadhyay, S., Coyle, E.J.: An Energy Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks. In: Proceedings of IEEE INFOCOM 2003, San Francisco, vol. 44(1), pp. 1–16 (April 2004)
2. Lin, C.R., Gerla, M.: Adaptive Clustering for Mobile Wireless Networks. Journal on Selected Areas in Communication 15, 1265–1275 (1997)
3. Ye, W., Heidemann, J., Estrin, D.: Medium Access Control With Coordinated Adaptive Sleeping for Wireless Sensor Networks. IEEE/ACM Transactions on Networking 12(3), 493–506 (2004)

4. Zheng, J.: A Comprehensive Performance Study of IEEE 802.15.4. IEEE Press Book, Los Alamitos (2004)
5. IEEE P802.15.4/D18, Draft Standard: Low Rate Wireless Personal Area Networks (February 2003)
6. Wang, C., Li, B., Li, L.: A new collision resolution mechanism to enhance the performance of IEEE 802.11 DCF. IEEE Trans. on Vehicular Technology 53(4), 1235–1246 (2004)
7. Tay, Y.C., Jamieson, K., Balakrishnan, H.: Collision-Minimizing CSMA and its Applications to Wireless Sensor Networks. IEEE J. Selected Areas in Communications 22(6), 1048–1057 (2004)
8. Chatterjee, M., Das, S.K., Turgut, D.: WCA: A Weighted Clustering Algorithm for Mobile Ad hoc Networks. Journal of Cluster Computing, Special issue on Mobile Ad hoc Networking 5, 193–204 (2002)
9. Misic, Maintaining Reliability Through Activity Management in 802.15.4 Sensor Networks, Quality of Service in Heterogeneous Wired/Wireless Networks (2005), Second International Conference (August 2005)
10. Neugebauer, M.: A new beacon order adaptation algorithm for IEEE 802.15.4 networks. In: Proceeedings of the Second European Wireless Sensor Networks (2005)
11. Mhatre, V., Rosenberg, C.: Design Guidelines for Wireless Sensor Networks: Communication, Clustering and Aggregation. Ad Hoc Networks Journal 2(1), 45–63 (2004)
12. IEEE, IEEE Standard for Information Technology– Telecommunications and Information Exchange between Systems –Local and metropolitan area networks –Specific Requirements – Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANS), IEEE Standard 802.15.4 (October 2003)
13. IEEE, IEEE Standard for Information Technology– Telecommunications and Information Exchange between Systems – Specific Requirements – Part 11: Wireless LAN MAC and PHY Specifications, IEEE Std 802.11-1999, IEEE, New York (1999)
14. Bandyopadhyay, S., Coyle, E.J.: Spatio-Temporal Sampling Rates and Energy Efficiency in Wireless Sensor Networks. In: Proceedings of IEEE INFOCOM 2004, Hong Kong (March 2004), and IEEE/ACM Transactions on Networking (January 2006) (to appear)
15. Bharghavan, V.: MACAW: A Media Access Protocol for Wireless LANS. In: Proceedings of SIGCOMM 1994, London, England, pp. 212–225 (August 1994)
16. Kwon, Y., Fang, Y., Latchmari, H.: A Novel MAC Protocol with Fast Collision Resolution for Wireless LANs. In: Proceedings of IEEE INFOCOM 2003, San Francisco (April 2003)

Performance Analysis of IEEE 802.11 in Multi-hop Wireless Networks

Lan Tien Nguyen[1], Razvan Beuran[2,1], and Yoichi Shinoda[1,2]

[1] Japan Advanced Institute of Science and Technology,
1-1 Asahidai, Nomi, Ishikawa, 923-1292 Japan
[2] National Institute of Information and Communication Technology
Hokuriku Research Center, 2-12 Asahidai, Nomi, Ishikawa, 923-1211 Japan
lannt@jaist.ac.jp

Abstract. Multi-hop wireless networks provide a quick and easy way for networking when we need a temporary network or when cabling is difficult. The 802.11 Medium Access Control (MAC) plays an important role in the achievable system performance. There have been many studies on analytic modeling of single-hop 802.11 wireless networks but only a few on the analysis of multi-hop wireless networks. Furthermore, the object of these researches is an homogeneous ad-hoc wireless networks; therefore they are not appropriate for a network with structure such as wireless mesh networks. This paper introduces an analytic model of throughput performance for the IEEE 802.11 multi-hop networks, which allows us to compute the achievable throughput on a given path in multi-hop wireless networks. The model shows that there is an optimal point at which throughput is maximized. Using this model and a Markov model for modeling the operation of the IEEE 802.11 DCF we can determine the amount of data that each node should inject to the network to get the best throughput performance.

1 Introduction

Multi-hop wireless networks provide a quick and easy way for networking when we need a temporary network or when cabling is difficult. The 802.11 Distributed Coordination Function, Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) based, is the most popular MAC protocol for wireless communication. The analysis in this paper is based on the operation of 802.11 DCF.

In wireless networks, the throughput that can be sent through a wireless link depends on various factors such as the distance between nodes, the transmission power, characteristics of environment like path loss, fading, noise, etc. Given the broadcast nature in wireless environment, the achievable throughput on a wireless link not only dependent on the operation data rate on that link, but also depends on number of nodes using the same radio channel within the two end-points carrier sense area. Considering a traffic flow sent from source node to destination node through other intermediate nodes, the middle nodes may have to contend with more nodes than the source node or the destination node does. Consequently, the source node can inject

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 326–337, 2007.

more data into the path than the amount that can be forwarded by middle nodes. This may lead to packet loss and low performance in wireless network since the cost to resend a packet is high and when injecting the amount of data larger than the one the channel can accepted makes the condition even worse and leads to a high packet loss. High packet loss in turn can trigger re-routing and make the network topology instable [1].

To date, there have been a lot of studies on analytic modeling of both single-hop 802.11 wireless networks [2], [3], [4], [5], [6] and multi-hop wireless networks [7], [8], [9] [10]. All of these models are assumed to use saturated traffic load (which mean a node always has a packet ready for transmission) except the model which is proposed in [10]. Other studies focus on the theoretical upper bound of throughput on an homogeneous ad-hoc network [11], [12] or are based on the assumption of global scheduling [13], [14], [15], which may not be a good assumption in a real wireless networks using IEEE 802.11.

In this paper, we introduce an analytic model of throughput performance for 802.11 multi-hop networks allowing us to compute the achievable throughput on a given path in multi-hop wireless networks. The model shows that there is an optimal point at that throughput is maximized. Based on this model and Markov model for modeling the operation of the IEEE 802.11 DCF we can determine the amount of data that each node should inject to the network to get the best throughput performance.

2 Related Work

Following the analytical methodology introduced for the analysis of ALOHA protocol and carrier sense multiple access [16], [17], a lot of analytical modeling of wireless MAC protocols has often focused on single-hop wireless networks. Although some models are proposed for multi-hop wireless networks [18], [19], they can't model the effect of binary exponential back-off scheme (BEB) which is the key to adjust transmission intervals in IEEE 802.11. Recently there are more researches on modeling IEEE 802.11 DCF in single-hop wireless networks [2],[3], [4], [5], [6]. With a Markov Chain model, the exponential back-off scheme was accurately modeled. However, these models can't be directly applied to multi-hop wireless networks due to the hidden node problem.

One of the first analytical models of IEEE 802.11 DCF for multi-hop wireless networks is proposed in [7]. The hidden node problem is taken into account, but transmission of all the nodes is assumed to follow a Poisson process which doesn't match the behavior of IEEE 802.11 binary exponential back-off scheme. In the model which is proposed by Wang and Garcia-Luna-Aceves in [8] the BEB scheme can't be captured effectively either because only a simple model for exponential back-off scheme is used.

Some recent works have solved the problem to model behavior of binary exponential back-off scheme [9], [10]. Carvalho and Garcia-Luna-Aceves introduced an analytical model to study operation of 802.11 DCF in multi-hop wireless networks [9]. The model takes into account the impact of both physical layer and network topology. However, the impact of hidden nodes is not considered. David Malone et al. proposed a model which is effective in capturing the binary exponential back-off scheme of

802.11 DCF in both saturated and non-saturated environment [10]. This model helps us in studying the relation between input load and output load of a node depending on several parameters of IEEE 802.11 DCF.

There are some research results on the capacity of general ad-hoc network [11], [12] and mesh network [13]. It was shown that for stationary networks, the capacity for each node decrease as $O(1/\sqrt{n})$; meanwhile for the mobile networks that can tolerate long delay, the capacity may remain constant. In mesh networks, the authors in [13] claimed that gateways are bottlenecks and the available capacity for each node reduces to $O(1/n)$, where n is the number of node associated with one gateway.

3 Network Model

We consider a multi-hop wireless network in which each node uses IEEE 802.11 DCF for medium access control and has only one radio interface operating on the same radio channel. In our model we also assume that only one transmission rate is used although 802.11 a/b/g standards support multiple transmission rates

At the MAC layer, the data payload is assumed to be 1024 bytes plus 34 bytes from MAC header. Request-to-Send (RTS) and Clear-to-Send (CTS) are assumed to be sent at the lowest data rate that is supported by physical layer.

At the application layer, we assume that there is single data flow and our goal is to present an analytical model to compute achievable throughput along a path. Because of having only one flow in the network so the model we use only consider intra-flow interference occurring for packets of the same flow transported over different wireless links. The inter-flow will be taken into account in our future model.

4 Throughput Analysis

The analytical model that we use to compute achievable throughput along a wireless multi-hop path is based on the model presented in [2] but it takes into account hidden nodes which strongly affects achievable throughput in multi-hop wireless networks.

Similar to [11] we define three radio ranges:

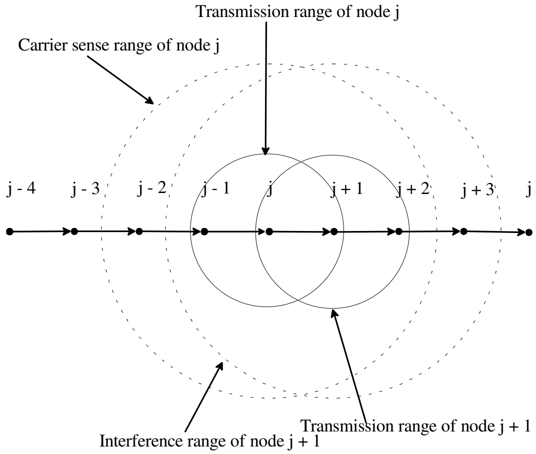
Transmission range (R_{tc}): represents the range in which a frame can be successfully received if there is no interference from other nodes. This value is determined by transmission power, receiver sensitivity and radio propagation properties. A node B is considered in transmission range of node A if packets come from node A are received at node B with power higher than minimum reception power of node B.

Carrier sense range (R_{cs}): represents the range in which a transmission can trigger carrier sense detection at radio interface of the node. This value is determined by receiver's sensitivity and also transmission power, radio propagation properties. A node B is considered in carrier sense range of node A if packets come from node A are received at node B with power higher than minimum detection power of node B.

Interference range (R_i): represents the range in which the station in receive mode will be interfered by other transmitter and thus suffers a loss.

Considering a transmission from node A to node B, the hidden nodes in this case are the nodes that are inside the interference range of node B (receiver) but outside both transmission range of node B and carrier sense range of node A (transmitter). Let's say C is a node in the hidden node set. Because C is outside the carrier sense range of node A so it is not aware of transmissions between A and B. Any transmission of C will corrupt the transmission between A and B. The RTS/CTS scheme can't solve hidden node problem, and this reason makes performance of IEEE 802.11 worse in multi-hop wireless networks. Our model will take into account the impact of the hidden node problem on the throughput of a given path.

We consider *n* fixed nodes and *n* - 1 nodes send traffic to the next nodes along that path as illustrated in the Figure 1. The following notations are used to denote subset of *n* nodes.



Transmission range of node j
Carrier sense range of node j
j - 4
j - 3
j - 2
j - 1
j
j + 1
j + 2
j + 3
j +
Transmission range of node j + 1
Interference range of node j + 1

Fig. 1. Illustration of the three different ranges

- C_j refers to subset of nodes within carrier sense range of node *j*.
- C_j^+ refers to the subset C_j plus node *j* itself.
- I_j refers to the subset of nodes within interference range of node *j*.
- T_j refers to the subset of node within transmission range of node *j*.

Let *S* to be normalized system throughput, which can be compute as fraction of time that channel is used to successfully transmit payload data. In a slot time channel can be idle, busy with a successful transmission or busy with a collision. In order to calculate the average length of a slot time we have to consider what can happen in a slot time or the probability that channel has one of three different states. Let *n* be maximum number of nodes in the interference range of nodes in the path, τ the probability that a node transmits in a random chosen slot time, and P_{tr} the probability that there is at least one transmission in the considered slot time. One time slot is denoted as σ. Because of *n* different nodes content on the channel, each node will transmit in a slot time with probability τ.

$$P_{tr} = 1 - (1 - \tau)^n \tag{1}$$

Consider a transmission between node *j* and node *j+1* and successful transmission need *k* time slot. Let P_s be the probability that the transmission is successful, which is given by probability that there is only the sender transmit at the time in C_j and none of nodes in $\{I_j\ U\ I_{j+1}\}/\{T_j\ U\ T_{j+1}\}$ transmit in *k* time slot. Let *h* be number of nodes in $\{I_j U\ I_{j+1}\}/\{T_j\ U\ T_{j+1}\}$, these nodes are called hidden nodes.

$$P_s = \frac{n\tau(1-\tau)^{n-1}(1-\tau)^{hk}}{P_{tr}} \tag{2}$$

We can express the achievable throughput *S* as the following ratio

$$S = \frac{E_p}{E_{sl}} \tag{3}$$

Where E_p is average amount of payload information is successfully transmitted in a slot time, E_{sl} being average length of a slot time.

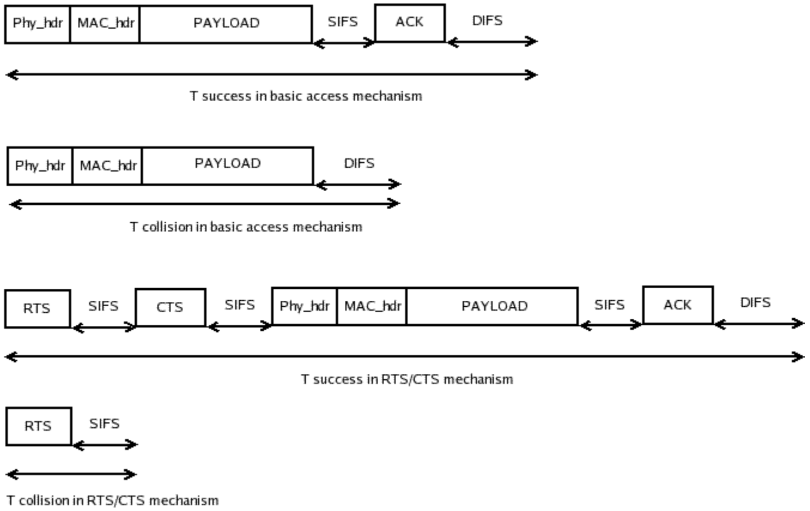

Phy_hdr
MAC_hdr
PAYLOAD
SIFS
ACK
DIFS
T success in basic access mechanism
Phy_hdr
MAC_hdr
PAYLOAD
DIFS
T collision in basic access mechanism
RTS
SIFS
CTS
SIFS
Phy_hdr
MAC_hdr
PAYLOAD
SIFS
ACK
DIFS
T success in RTS/CTS mechanism
RTS
SIFS
T collision in RTS/CTS mechanism

Fig. 2. Success time and collision time with basic access and RTS/CTS mechanism

Let *E(P)* be average packet payload size. Since a packet can be transmitted successfully in a slot time with probability $P_{tr}.P_s$, we may have

$$E_p = P_{tr} P_s E(P) \tag{4}$$

The average length of a slot time is obtained considering that: a slot time is idle with probability $1\text{-}P_{tr}$; contain a successful transmission with probability $P_{tr}.P_s$ and contains a collision with probability $P_{tr}(1\text{-}P_s)$. Thus equation (3) can be expressed as

$$S = \frac{(1/n)P_{tr}P_s E[P]}{(1-P_{tr})\sigma + P_{tr}P_sT_s + P_{tr}(1-P_s)T_c} \tag{5}$$

Where T_s is the average time that the channel is sensed busy by a successful transmission, T_c being average time that channel is sensed busy by a collision and σ is duration of a slot time. The value of T_s and T_c depends on the access mechanism used.

Firstly, we consider the system uses the basic access mechanism. Denoting δ as propagation delay, from Figure 2 we can obtain the value of T_s^{bas} and T_c^{bas} as follows

$$T_s^{bas} = Phy\_hdr + MAC\_hdr + E[P] + SIFS + \delta + ACK + DIFS + \delta \tag{6}$$

$$T_c^{bas} = Phy\_hdr + MAC\_hdr + E[P^*] + DIFS + \delta \tag{7}$$

where *E[P*]* is the average length of longest packet involve in the collision. In our model we assume that all packets have the same payload length so that *E[P*] = E[P] = P*

Secondly, considering the system in which each packet is transmitted by using RTS/CTS mechanism T_s and T_c can be computed by the following equations:

$$\begin{aligned} T_s^{RTS} &= RTS + SIFS + \delta + CTS + SIFS + \delta + Phy\_hdr \\ &+ MAC\_hdr + E[P] + SIFS + \delta + ACK + DIFS + \delta \end{aligned} \tag{8}$$

$$T_c^{RTS} = RTS + DIFS + \delta \tag{9}$$

In order to see the relation between S and τ we rearrange equation (5) as

$$S = \frac{(1/n)E[P]}{T_s - T_c + \dfrac{\sigma}{P_{tr}P_s}\left[P_{tr}(T_c^* - 1) + 1\right]} \tag{10}$$

where $T_c^* = \dfrac{T_c}{\sigma}$.

As T_s, T_c, n, $E[P]$ and σ are constants and let us call

$$F(\tau) = \frac{1}{P_{tr}P_s}\left[P_{tr}(T_c^* - 1) + 1\right] \tag{11}$$

From (1) (2) and (11)

$$F(\tau) = \frac{1}{n}\left[\frac{T_c^*}{\tau(1-\tau)^{n+hk-1}} - \frac{T_c^* - 1}{\tau(1-\tau)^{hk-1}}\right] \tag{12}$$

By analyzing the relation between F and τ we can get the relation between S and τ. Taking the derivative of (12) with respect to τ.

$$\frac{dF}{d\tau} = \frac{(T_c^* - 1)(1-\tau)^n + T_c^*\tau(n+hk) - T_c^*}{n\tau^2(1-\tau)^{n+hk}} \tag{13}$$

It is easy to see that

$$\lim_{\tau \to 0^+} \frac{dF}{d\tau} < 0 \tag{14}$$

$$\lim_{\tau -> 1^-} \frac{dF}{d\tau} > 0 \tag{15}$$

From (14) and (15) there must be a value of τ in the range [0:1] at that the value of F is minimum and hence it maximizes the value of S. It is possible to assume that $\tau \ll 1$. That assumption comes from the mechanism of the IEEE 802.11 DCF standards [12]. According to [12] the minimum value of contention window (CW_0) is 32 so the value of τ should be smaller than $1/CW_0$ (0.0321) and our assumption is valid. To find that value of τ under the condition $\tau \ll 1$ we can approximate

$$(1-\tau)^n \approx 1 - n\tau + \frac{n(n-1)}{2}\tau^2 \tag{16}$$

Making equation $\dfrac{dF}{d\tau} = 0$ to be quadratic equation and then we can solve it.

Table 1. System Parameters

Packet Payload	1024 bytes
MAC header	272 bits
Physical header	192 bits
ACK	112 bits + Physical header
RTS	160 bits + Physical header
CTS	112 bits + Physical header
Channel Bit Rate	1 Mbit/s
Propagation delay	1 μs
Slot Time	20 μs
SIFS	10 μs
DIFS	50 μs

Figure 6 shows the theoretical maximum throughput that can achieve with DCF with the RTS/CTS mechanism. The parameters used to compute numerical results are summarized in Table 1.

5 Simulation and Analysis Results

In this section some results of both analysis and simulation are presented for evaluating the effectiveness of the proposed analytical model. For simulation we use ns-2.27 [22], with CMU Monarch Project wireless and mobile ns-2 extensions [23]. The network topology used includes 100 nodes, which are putted in a line. The distance between two nodes is changed from 200 m to 54 m to vary the number of nodes in the interference range. The transmission range and carrier sense range are set to be 250 m and 550 m, respectively. The data rate used in all simulations is 1 Mbps and the other parameters are set according to the Table 1. The traffic source for a node will send data at several constant rates; these rates are set to one of the following rate 20, 30, 40, 50, 70, 90, 120, 150, 200 Kbps for monitoring the changes of throughput at each node.

The simulation results in Figure 3, 4 and 5 show that when the input traffic load increases 802.11 multi-hop wireless network will get to saturation status but with higher number of node in the interference range the throughput will be decreased quickly because of collisions.

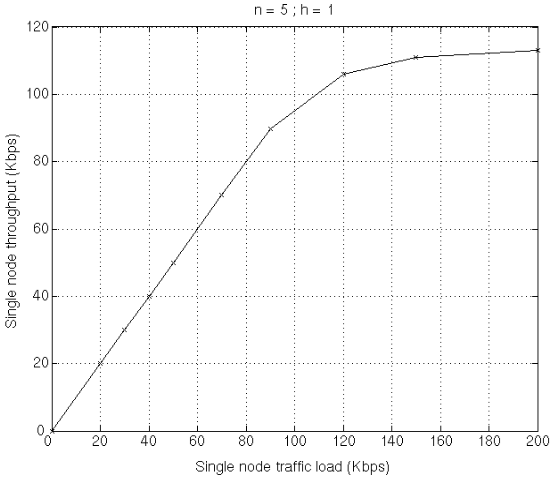


Fig. 3. Single node throughput versus single node traffic load ($n = 5$; $h = 1$)

In the analytical model, the input traffic load is represented by transmission probability τ, where τ is the probability that a node transmits in a random chosen slot time. The value of τ obviously depends on the node's input traffic load, traffic of other

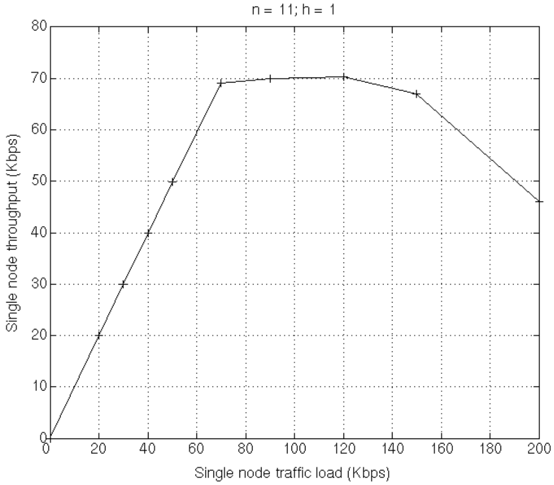


Fig. 4. Single node throughput versus single node traffic load ($n = 11$; $h = 1$)

nodes in its carrier sense range and the mechanism is used to access media (IEEE 802.11 DCF in this case). By using a Markov model, the relation between τ and input traffic load can be obtain from a research of K. Duffy et al [10].

$$\tau = \frac{1}{\eta}\left(\frac{q^2 W_0}{(1-q)(1-p)(1-(1-q)^{W_0})} - \frac{q^2(1-p)}{1-q}\right) \tag{17}$$

where

$$\begin{aligned}\eta &= \frac{qW_0}{1-(1-q)^{W_0}} + \frac{qW_0(qW_0+3q-2)}{2(1-q)(1-(1-q)^{W_0})} \\ &+ (1-q) + \frac{q(W_0+1)(p(1-q)-q(1-p)^2)}{2(1-q)} \\ &+ \frac{pq^2}{2(1-q)(1-p)}\left(\frac{W_0}{1-(1-q)^{W_0}} - (1-p)^2\right) \\ &\left(\frac{2W_0(1-p-p(2p)^{M-1})}{(1-2p)} + 1\right)\end{aligned}$$

Here, p is the probability that a node senses the channel busy on an attempted transmission, q being the probability that the node's buffer has packets waiting for transmission, W_0 being the minimum contention window of the node and W_0^M being the node's maximum contention window size. M is the maximum back-off stage.

On other hand, based on the nature of wireless environment we have a relation between τ and p

$$1 - p = (1-\tau)^{n-1} \tag{18}$$

where n is number of node in the interference range of the node including itself.

Based on the assumption that we can compute the value of q from given input traffic load of the node, by solving equations (17) and (18) we can find the value of p and τ at that the node operates. Getting the value of τ by doing so is somehow difficult. However, with the given network topology, the analytical model can predict the maximum throughput of a node or the input traffic load can be used to saturate the network.

Figures 3, 4, and 5 present the simulation results with different network conditions while Figure 6 shows the analytical results. It is can be seen that the maximum values of throughput from simulation results are reasonably consistent with the maximum throughputs from analytical results. The maximum values of throughput in figures 3, 4, and 5 are 114 kbps, 70 kbps, 36 kbps while the analytical model gives us the values of 106 kbps, 59 kbps, 31 kbps respectively. Thus we can say that the analytical model can be applied to predict the saturation throughput on a given path of 802.11 multi-hop wireless networks and consequently the optimum input traffic load is obtained.

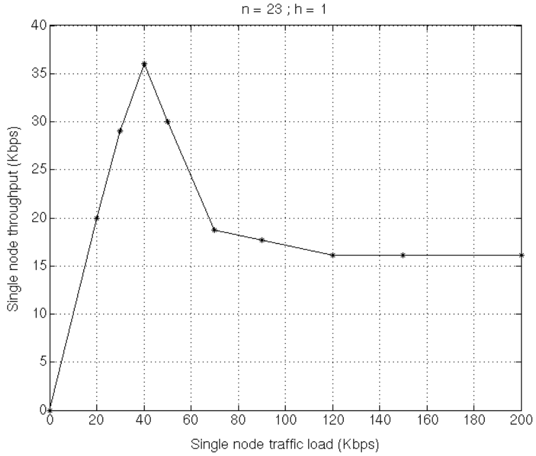


Fig. 5. Single node throughput versus single node traffic load (n = 23; h = 1)

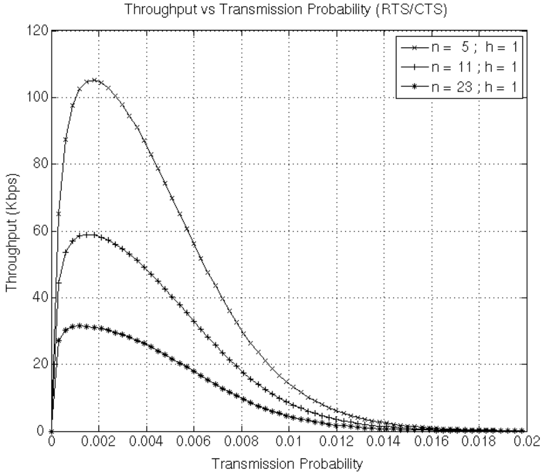


Fig. 6. Single node throughput versus transmission probability τ, obtained by the analytical model

6 Conclusions

In this paper we propose an analytical model for analyzing the throughput performance of IEEE 802.11 multihop wireless networks. Comparison with simulation results shows that the model is successful to estimate the saturation throughput on a given path in the multihop wireless network. The model also allows us to understand how the interference range and hidden node affect to throughput performance or say in other way the impact of physical conditions to MAC performance. By using this model performance of MAC protocol in IEEE 802.11 multihop wireless network can be studied more effectively. For the future work, we will study the issue of improving throughput performance in IEEE 802.11 wireless mesh network and in particular, we will focus on routing protocol and admission control mechanism for that network. We will also need a better model which allows us to predict per hop throughput as well as delay time and jitter with inter-flow interference and multiple transmission rates will be taken into account.

Acknowledgements. The financial support from the 21st century COE program "Verifiable and Evolvable e-Society", Japan Advanced Institute of Science and Technology, is gratefully acknowledged.

References

1. Ng, P.C., Liew, S.C.: Re-routing Instability in IEEE 802.11 Multi –hop Ad hoc Networks. In: IEEE WLN 2004, Tampa, USA (November 2004)
2. Bianchi, G.: Performance analysis of the IEEE 802.11 distributed coordination function. IEEE Journal on Selected Areas in Communications 18(3), 535–547 (2000)
3. Manshaei, M.H., Cantieni, G.R., Barakat, C., Turletti, T.: Performance Analysis of the IEEE 802.11 MAC and Physical Layer Protocol. In: WOWMOM 2005, pp. 88–97 (2005)
4. Foh, C.H., Zukerman, M.: Performance Analysis of the IEEE 802.11 MAC Protocol. In: EW2002 Proceedings (2002)
5. Xiao, Y., Rosdahl, J.: Performance Analysis and Enhancement for the Current and Future IEEE 802.11 MAC Protocols. ACM SIGMOBILE Mobile Computing and Communications Review (MC2R), special issue on Wireless Home Networks 7(2), 6–19 (2003)
6. Cali, F., Conti, M., Gregori, E.: Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit. IEEE/ACM Transactions on Networking 8(6), 785–799 (2000)
7. Chhaya, H., Gupta, S.: Performance modeling of asynchronous data transfer methods of IEEE 802.11 MAC protocol. Wireless Networks 3, 217–234 (1997)
8. Wang, Y., Garcia-Luna-Aceves, J.J.: Performance of collision avoidance protocols in single-channel ad hoc networks. In: Proc. of ICNP, pp. 184–190 (November 2002)
9. Carvalho, M., Aceves, J.: Scalable model for channel access protocols in multihop ad hoc networks. In: ACM Mobicom 2004 (September 2004)
10. Duffy, K., Malone, D., Leith, D.J.: Modeling the 802.11 Distributed Coordination Function in non-saturated conditions. IEEE Communications Letters 9(8), 715–717 (2005)
11. Gupta, P., Kumar, P.R.: The Capacity of Wireless Networks. IEEE Trans. Inform. Theory 46(2), 388–404 (2000)
12. Li, J., Blake, C., et al.: Capacity of Ad Hoc Wireless Networks. In: ACM MobiCom 2001, Rome, Italy (July 2001)
13. Jangeun, J., Sichitiu, M.L.: The nomial capacity of wireless mesh networks. IEEE Wireless Communications, 8–14 (October 2003)
14. Jain, K., et al.: Impact of Interference on Multi-hop Wireless Network Performance. In: ACM MobiCom 2003, San Diego, USA (September 2003)
15. Kodialam, M., Nandagopal, T.: Characterizing Achievable Rates in Multi-hop Wireless Networks: The Joint Routing and Scheduling Problem. In: ACM MobiCom 2003, San Diego, USA (September 2003)

16. Kleinrock, L., Tobagi, F.A.: Packet switching in radio channels: Part I - carrier sense multiple-access modes and their throughput-delay characteristics. IEEE Transactions on Communications 23(12), 1400–1416 (1975)
17. Tobagi, F.A., Kleinrock, L.: Packet switching in radio channels: Part II - the hidden terminal problem in carrier sense multiple-access modes and the busy-tone solution. IEEE Transactions on Communications 23(12), 1417–1433 (1975)
18. Tobagi, F.A.: Analysis of a two-hop centralized packet radio network - part II: Carrier sense multiple access. IEEE Transactions on Communications 28(2), 208–216 (1980)
19. Tobagi, F.A.: Analysis of a two-hop centralized packet radio network - part II: Carrier sense multiple access. IEEE Transactions on Communications 28(2), 208–216 (1980)
20. Xu, K., Gerla, M., Bae, S.: How effective is the IEEE 802.11 RTS/CTS handshake in ad hoc networks. In: Proc. of GLOBECOM 2002 (2002)
21. IEEE standards for wireless LAN medium access control (MAC) and physical layer (PHY) specifications (1999)
22. The network simulator - ns2, http://www.isi.edu/nsnam/ns/
23. CMU Monarch Project Extensions to NS2, http://www.monarch.cs.cmu.edu/cmu-ns.html

ASDP: An Action-Based Service Discovery Protocol Using Ant Colony Algorithm in Wireless Sensor Networks

Hongwei Huo, Deyun Gao, Yanchao Niu, and Shuai Gao

School of Electronics and Information Engineering, Beijing Jiaotong University,
Beijing, 100044, China
hwhuo@ieee.org, gniux819@gmail.com, {gaody,shgao}@bjtu.edu.cn

Abstract. In large-scale wireless sensor networks, efficient service discovery and data transmission mechanisms are both essential and challenging. Ant colony algorithm which has been used to resolve routing, localization and object tracing issues in mobile ad hoc and sensor networks provide a valuable solution for this problem. In this paper, we describe a novel scalable Action-based Service Discovery Protocol (ASDP) using ant colony algorithm in wireless sensor networks. ADSP can abstract the semantics information from the data via the nodes or user operation and map them into six different action sets. Then it adjusts the related parameters to satisfy with service and transmission requirements from different kinds of actions. We evaluate it against other approaches to identify its merits and limitations. The simulation results show that ASDP can maximize the network utilization. Farther experiments indicate it scales to large number of nodes.

1 Introduction

Typically, services are applications offered by service providers which users can interact with. A service discovery protocol allows service providers to advertise their services, and also allows users to automatically discover such services [4][5]. Wireless Sensor Networks (WSNs), as an emerging technology, open a wide perspective for future applications in ubiquitous computing. It is not unpredictable that large-scale sensor networks eventually will become one of the most important infrastructures in our life, which can be integrated into homes, offices, public places, battle fields and all kinds of natural environments, where service discovery will be revered significantly. The concept of service can be extended as either hardware or software resources, accessible through the network and capable of interacting with each other by sending and receiving information and data. And the service discovery can be comprehended as nodes are aware of their own capabilities, automatically locate network services according to the semantic attributes and to advertise their own capabilities to the rest of the network, by which they can cooperate with other nodes in the network, for the purpose of providing networking and system services such as user querying or data recording [2] [3].

For efficient service discovery, the main challenges in wireless sensor networks are the limited power, computational capacities, memory and prone to failures. Some

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 338–349, 2007.

applications involve mobility and large number of nodes. In WSNs, the network actions consists of the task sent to the nodes and the data recoded by nodes, thus communication is application-specific and data-centric. Typical approaches for service discovery and followed data transmission rely on either flooding or centralized storage. Both may cause efficiency, scalability and robustness problems. In this paper, we describe a novel scalable, efficient and robust Action-based Service Discovery Protocol-ASDP using Ant colony algorithm, which has been widely used to resolve routing, localization and object tracing problems in mobile ad hoc and sensor networks, to maximize the network utilization. Considering the special requirement of services discovery in wireless sensor networks, we select the basic application scenarios, which consist of user query and data recording, and create a scheme to abstract the semantics information from the data generation via the nodes or user operations via the sinks. All these information are mapped in to six different basic network action sets: emergency querying, emergency report, normal querying, normal report, emergency cooperation and normal cooperation. ASDP selects different parameter combination of the ant-based algorithm to achieve the service requirement separately. We ran extensive simulations to investigate our approach, and evaluated it against other approaches to identify its merits and limitations. The results show that ASDP is scalable to large number of nodes and is highly efficient with action's changes.

The rest of this paper is organized as follows. Section 2 describes the service discovery framework in wireless sensor networks. In section 3, we present an action based ant colony optimization algorithm which is used as the soul of our work. Section 4 explains our service discovery protocol design, followed in Section 5 by simulation and performance evaluation results. We discuss related work in section 6. Finally, in section 7 we give out some conclusions.

2 Action-Based Service Discovery in WSNs

We consider the solutions for service discovery depend on the application-specific and data-centric network action, which consists of the task sent to nodes and the data recoded by nodes[5][6]. According to different user requirements, Service Discovery protocol should perform different actions: service query, service match and service reply, which will perform different schemes. In the basic application scenarios of WSNs, the sensor nodes are deployed randomly or regularly to perform sensing, caching, processing, and wireless transmission of sensor data via self-organizing. Users can pre-configure the sensor nodes to record sensory data and to report them periodically or according to the user's request[10]. Figure.1 depicts the 3 basic service discovery types in detail. In this paper, we abstract this process into six basic actions: emergency querying (EQ), emergency report(ER), normal querying(NQ), normal report(NR), emergency cooperation(EC) and normal cooperation(NC). Emergency querying is an abstraction of the querying by users via one or more sink nodes to obtain data from WSNs within an extremely short time, while normal querying is much like emergency querying but without time limitation. Emergency report means that the nodes should request the location of the sink nodes and report some emergency data to these nodes according to the pre-configuration. Normal report means that the nodes report regular data to these nodes periodic or on demand. To facilitate the cooperation of sensor nodes to support the reconfiguration capability or to achieve a common goal of the network, sensor nodes should also request services from each other, thus we define another two basic actions: emergency cooperation for real-time scenarios and normal cooperation for non-emergency scenarios. A good example is the monitoring of emergency or disaster scenarios, such as floods, fires, earthquakes, and landslides. Sensor nodes used to collect and transmit data can fail suddenly and unpredictably as they may melt, corrode or get smashed. This sudden, unpredictable failure is especially troubling because of the potential loss of data collected by the sensor nodes. Thus data backup between sensor nodes is an important mission for this kind of WSNs. The nodes should request the data backup service to other nodes with potential storage space[11]. This process is a kind of emergency cooperation action.

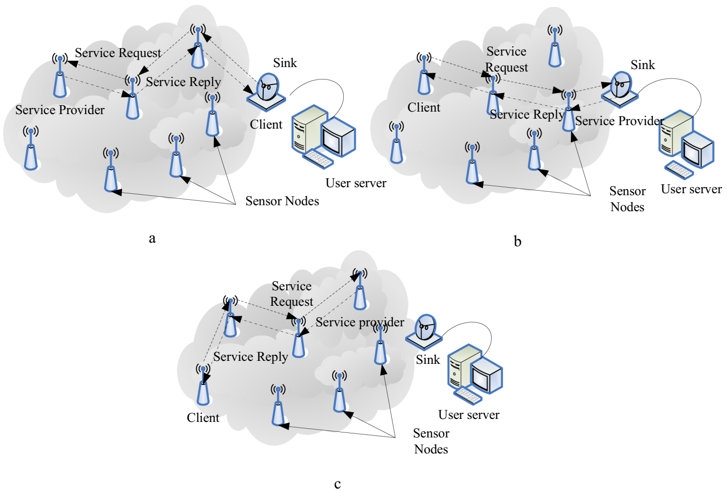
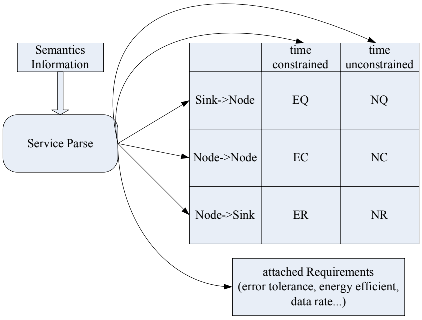


Fig. 1. Three basic service discovery : a)sink nodes as client and sensor nodes as provider. b) sensor nodes as client and sink nodes as provider. c) both client and provider are sensor nodes.

All the services in WSNs should consult one or more of these actions and assimilate them as their properties according to the semantics information. In fact how to describe the service and how to map semantics information into network action is out of our scope. In this paper we mainly pay attention to how to fulfill the communication request of the service discovery in each action. However, we also give a brief introduction. Figure.2 summarizes the approach to map the semantics information into basic actions. We hold a service parser as an interface to parse the semantics information into different attributes, such as session origination, session destination, time constraints, energy efficiency, etc… When a session from top layer is received, service parser will check these attributes and classify this session into its basic action to fulfill the requirement of the session.

Service discovery might be affected by architecture, available resources, incorporated routing and communication mechanisms[1].But typical approaches for service discovery and followed data transmission rely on either flooding or centralized storage, which may cause efficiency, scalability and robustness problems. These approaches can not fulfill the request of all actions of the network concurrently. Many biological systems exhibit properties of self-organization and emergence, which are inherently adaptive, but there is no centralized control. Clearly such systems are somewhat parallel with WSNs, which likewise lack central controllers and must adapt to prevailing conditions[6].


Semantics Information
Service Parse
attached Requirements (error tolerance, energy efficient, data rate...)

	time constrained	time unconstrained
Sink->Node	EQ	NQ
Node->Node	EC	NC
Node->Sink	ER	NR

Fig. 2. Approach to map the semantics information into basic actions

3 Ant Colony Optimization Algorithm

The ant can seek path between the nest and multiple food sources. They accomplish the mission with great efficiency. When the environment changes, ants can also quickly discover new routes. Since Service Discovery is a process to find the best service and steer the service request from the client to the service provider, and then transmit the response back to the client. In dynamic and stochastic networks, such as WSNs, to search the best service and establish a proper route between them is a NP-hard Problem[14]. The idea of ASDP is inspired by the process used by ant colony to discover food resources, in which individual ants are capable of finding the best or relative Optimized path to the food sources or the nest without global administration. It is desirable for our Service discovery protocols to achieve optimization performance for all actions. In this section, we will introduce the Action-based Ant Colony Optimization Algorithm in detail.

3.1 Assumptions and Definitions

We assume that the network is represented by a weighted graph $G = (E, V)$, where V is the set of nodes and E is the set of edges. In WSNs, since sensor nodes may join or fail randomly, G can be considered to be a dynamic graph. Let Si denotes the set of Sink nodes, thus $N = V - Si$ denotes the set of sensor nodes. If node i and node j can communication directly via wireless radio, the link between *i* and *j* can be represented by $e^{i,j}$,where, $e^{i,j} \in E$. We give some assumptions and definitions:

In this paper, we assume that 1) all the communication links are symmetric, despite the link are asymmetrical in many practical communication systems. In fact, the symmetrical link can be assured through the MAC layer protocol.2) the local clock of the node is synchronized or approximately synchronized. This is very important, because the delay between nodes need to calculate via local clock.3) all of sensor nodes here have the same initial energy, denoted as E_0 .4) the localization coordinates of sensor nodes are known and the nodes were located at the same horizontal surface. Thus, we can use the 2-D Euclidean distance formula to calculate the distance of two sensor nodes. This assumption is crucially useful in simulation experiments.

Definition 1. Service Requirement. *The demand for services of each network action is different。 To facilitate the description of our algorithm, we define two basic* ***Service Requirements*** *according to the different network actions :* ***Delay*** *and* ***Potential Energy***。

Delay. *We use* $D_L^{i,j}$ *to represent the end to end delay between two nodes i and j. For a service S, we denote the path between the Client (denoted as 1) and Service Provider (denoted as m)* $P(S)$ *.Thus the delay for a service path is* $D_L(P(S)) = \sum_{i=1}^{m-1} D_L^{i,i+1}$.

Potential Energy. *We use* $E_D^{i,j}$ *represent the energy consumption caused by the communication between nodes i and j .For* $P(S)$ *which is defined above, the* ***energy consumption*** *of service path(****Potential Energy****) is* $E_D(P(\overline{S})) = \sum_{i=1}^{m} E_D^{i,i+1}$.

Definition 2. Target Value Function. *In the case of EQ, EC and ER, the we mainly consider delay, rather than energy. Whereas, in the case of NQ, NC and NR, we mainly consider energy, and delay is not an obvious concern. Therefore, we can set and adjust the action chosen parameter* $\vec{\gamma}$ *neatly according to the semantics describe of service request to reflect the specific service requirement. Where,* $\vec{\gamma}$ *is a 2-D vector,* $\vec{\gamma} = [\gamma_D, \gamma_E]$, $\gamma_D, \gamma_E \in [0,1]$ *.And* $\vec{\gamma}$ *can also be seen as the weight parameters of the* ***Service Requirement***

For a Service Discovery process, we define a Target Value Function, f :

$$f(P(S)) = \vec{\gamma} \times \begin{bmatrix} \omega_1 \times (\overline{D_{\text{threshold}}} / 2 - D_L(P(S))) \\ \omega_2 \times (| P(s) | E_0 - E_D(P(S))) \end{bmatrix} \tag{1}$$

For each Service, we always choose the highest value of f as the best service and the path of this service as the service reply route. Where, $\overline{D_{\text{threshold}}}$ represent the Maximum allowable delay of the delay threshold. n. $| P(s) |= m$ represents the total hop count of the path.

At this point, the problems of action based service discovery can be described as: for a service S, try to find a subset of E and one of its edge sets (a subset of V) in graph G to create a path set P(S), which makes numerical value of $f(P(S))$ as great as possible.

3.2 Basic Procedures

As mentioned above, the ant colony optimization algorithms have been inspired by the behavior of the real ant colony. In order to find the optimum solution, we generate several artificial ants, which search the solution space. The probabilistic movement of the ant allows the ant to explore new paths and find the proper service provider. We define three types of ants in this algorithm: forward ant(***Fa***), backward ant(***Ba***), and service ant(***Sa***). ***Fa*** is generated from the client to explore a path to a proper service provider. ***Ba*** is generated from the service provider and route back to the client. ***Sa*** is also generated from service provider to publish its services. High level flow chart for the action based ant colony optimization algorithm is depicted in Fig.3.

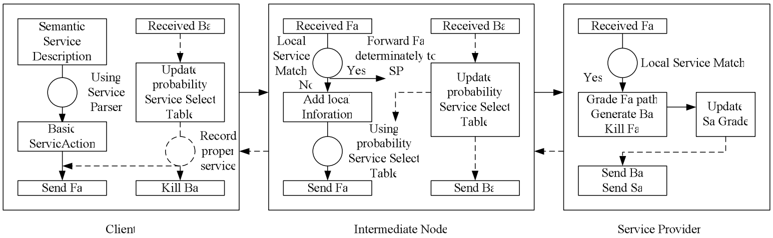


Fig. 3. High level flow chart describing basic procedures

When a client (either node or a sink) in the WSNs wants to find a service and /or maintain a path from the client to the service provider, it sends Fa to search for the proper service provider. For the service request is parsed from Semantic Service Description, ***Fa*** carried the action chosen parameter mapped via Basic Action to adjust the service discovery process. The generation rate of the ***Fa***s are determined by the network dynamics and the required ability to quickly respond to changes of the WSN. A Fa moves in WSNs to search the service provider using the Probability Service Select Table (PSST). PSST is maintained in each node according to which, the ***Fa*** probabilistically selects the next hop in its searching path. Figure.4 shows the structure of the PSST, where $\Pr_{i,k,j}$ (can also be marked as $\Pr_{k,j}$ in some obvious scenarios) denotes the probability of current ***Fa*** to achieve the Service Provider j via the next hop k which is a neighbor of current node i. N_i denotes the number of the valuable neighbors of current Node i, and N_{sp} is the number of the valuable service providers. $\Pr_{i,k,j}$ is calculated as following:

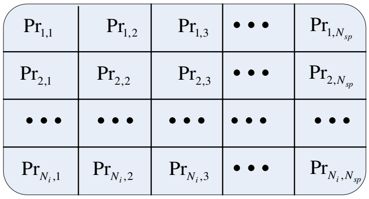
$$\Pr_{i,k,j} = \begin{cases} \dfrac{(\tau_{i,k})^{\alpha}(\eta_{i,k})^{\beta}}{\sum_{u\in N_i}(\tau_{i,u})^{\alpha}(\eta_{i,u})^{\beta}}, & v_k \notin tabu(Fa) \\ 0 & , v_k \in tabu(Fa) \end{cases} \tag{2}$$

Where, *tabu()* is used to represent the set of nodes that the ant has already passed. $\tau_{i,k}$ represents the current pheromone value on link $e^{i,k}$,which will be refreshed by

Bas. $\eta_{i,k}$ is the local heuristic gene, which represent the expectation to choose a proper neighbor. Here, we define $\eta_{i,k}$ as following:

$$\eta_{i,k} = \gamma_D \omega_1 (1 - 2D_L^{i,k} / D_{threshold}) + \gamma_E \omega_2 (E_D^{i,k} / E_0) \tag{3}$$

Thus, $\eta_{i,k}$ lay out that we always expect to choose the neighbor with high Energy Potential and low Delay links. α and β are two parameters to reflect the importance of current pheromone value and the local heuristic gene.



$\mathrm{Pr}_{1,1}$	$\mathrm{Pr}_{1,2}$	$\mathrm{Pr}_{1,3}$	• • •	$\mathrm{Pr}_{1,N_{sp}}$
$\mathrm{Pr}_{2,1}$	$\mathrm{Pr}_{2,2}$	$\mathrm{Pr}_{2,3}$	• • •	$\mathrm{Pr}_{2,N_{sp}}$
• • •	• • •	• • •	• • •	• • •
$\mathrm{Pr}_{N_i,1}$	$\mathrm{Pr}_{N_i,2}$	$\mathrm{Pr}_{N_i,3}$	• • •	$\mathrm{Pr}_{N_i,N_{sp}}$

Fig. 4. Probability Service Select Table of Node i

Fas collect the paths' information and intermediate nodes' local information as they travel. When a node received a ***Fa***, it first checks if the service requested by the ***Fa*** exists in its local service cache. If there is no local service match, ***Fa*** will be forwarded according current node's PSST. If there is local service match, the current node will differentiate whether the service is provided by itself or other nodes. If the service is not provided by itself, the node will forward the ***Fa*** to the Service provider determinately. Otherwise, it will grade the path information carried by the ***Fa*** and record this information. Then the ***Fa*** will be killed and a Ba will be generated and sent in the reverse path of its corresponding ***Fa*** with the grade φ, all ids of the intermediate nodes visited by ***Fa***. φ is the parameter which represent the ability of a service provider. Here, we define $\varphi = E_{sp} / E_0$, E_{sp} represents the current energy of service provider. That means the energy condition is the main factors to be considered for the ability of service provider.

As ***Fa*** move in the reverse path, the intermediate nodes will modify their pheromone value for the corresponding service and update the PSST. Here, $\tau_{i,k}$ will be updated according to the global information carried by the ***Ba***. When a ***Ba*** is received, the intermediate node collects the grade φ and calculate new $\tau_{i,k}$, using $\tau_{i,k}(n+1) \leftarrow \varphi \times \tau_{i,k}(n) + \Delta\tau_{i,k}$. Where,

$$\Delta\tau_{i,k} = \frac{\gamma_D \omega_0 (\overline{D_{threshold}} / 2 - D_L(P(s))) + \gamma_E \omega_1 (|P(s)| E_0 - E_D(P(s)))}{|P(s)|} \tag{4}$$

Finally, the client will receive the Ba, update its PSST and record the proper service. Then the Ba will be killed. We defined the initialized $\tau_{i,k}$ and $\Delta\tau_{i,k}$ randomly.

4 Special Consideration for ASDP Design

We noted that in massive sensor networks, the size of ***Sa*** and ***Ba*** will extend significantly. The ant will be so big that it may unfeasible to send the ant through the network. In this scenario, we can modify our packet format using the approach proposed in [15] and just maintain 3-most current hops in our ants, and the rest information can be stored in intermediate nodes.

When a service is deployed, it usually sends its service description to nearby nodes, which can increase the chance of its discovery by a matching service request. In our algorithms we use ***Sas*** to achieve this goal. When a service is deployed, it doesn't send any ***Sas*** until the first service request arrives. When the first service request for this service is arrived, a ***Sa*** with service description will be sent to all its neighbors with the value of TTL set as 1. If the Service provider received n service request from different clients during a fixed time interval Δt , it will send another ***Sa*** with a value of TTL n-1 at the end of this time interval. If the received service request came from the same client, it will send another ***Sa*** with a value of TTL 1. Else, it will not send anything about this service. In each neighbor of this service provider, the ***Sa*** will be killed and the information will be deleted if it can not receive a new ***Sa*** in a fixed time interval Δt. Then during the time interval, if another ***Sa*** with TTL more than 1 is received, it will update the ***Sa*** information and forward the ***Sa*** to its other neighbors with the TTL decreased by 1. This scheme means a popular service will have more chance of being discovered than a less popular one.

5 Simulation and Performance Evaluation

We implemented the ASDP using C++, and verified the performance of the proposed protocol through extensive simulation experiment. For executing comparison, we also implemented Flooding and HESED [22] in the same environment.

We first evaluate the Average Time Delay to match a service and Average Energy Consumption to match a service, using ASDP, HESED and Random Search(RSD) separately. In this pattern, we deployed only one service node in an $1000m \times 300m$ area. We increased the number of nodes from 20 to 100 to evaluate the performance of ASDP in small WSNs. The communication radius of them are 200m. In all experiments, the $D_{\text{threshold}} = 5s$. And we balance delay and energy consumption here. Figure.5 and Figure.6 illustrate the result of this experiment. We also evaluated the performance of ASDP in large scale WSNs. Figure.7, 8 show the results of this experiment. The results show that the delay of ASDP and HESED are nearly same, but ASDP is more energy efficient. Though RSP is more energy efficient than ASDP and HESED, the long time delay may be insufferable for most of the applications.

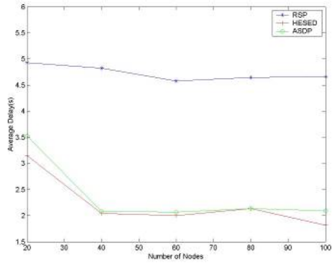


Fig. 5. Average Delay in small WSNs

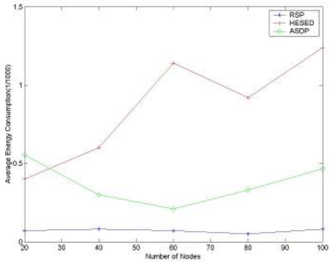


Fig. 6. Energy Consumption in small WSNs

As action based Service discovery protocol, we had to evaluate ASDP in different actions. Firstly, we imagine that there are more than one service in our WSNs such as NQ or EQ. We deployed 10 service nodes randomly and tried to match the best service in 5 seconds. We compared the response of HESED and ASDP. Figure.9 and Figure.10 show the total time and total energy to match the best service. The results indicate that HESED can match the best service quickly with more energy consumption, while ASDP has the opposite result. We take notice of the process of matching the best service, ASDP can finish this distributed rather than determined by client itself.

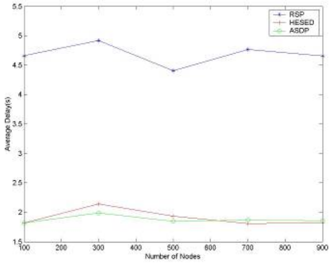


Fig. 7. Average Delay in Large Scale WSNs

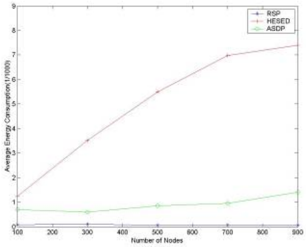


Fig. 8. Energy Consumption in large scale WSNs

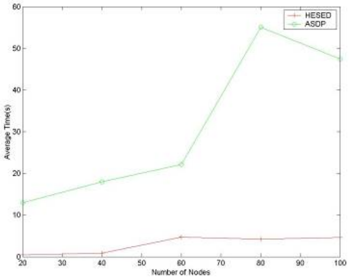


Fig. 9. Time of the best service matching

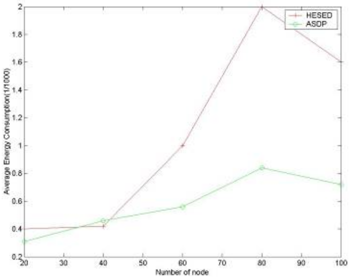


Fig. 10. Energy of the best service matching

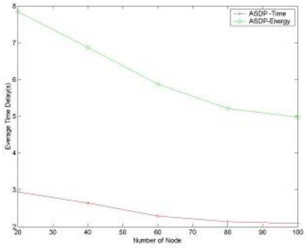


Fig. 11. Average Delay of different actions

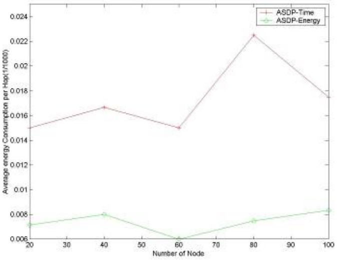


Fig. 12. Energy consumption of different actions

Secondly, we evaluate the affection of the weight parameters of the Service Requirement to ASDP. We set $\vec{\gamma}_1 = [50,0.5]$ and $\vec{\gamma}_2 = [0.5,50]$ to evaluate the action of ASDP. $\vec{\gamma}_1$ considers delay is more important and $\vec{\gamma}_2$ considers energy is more important. We showed the result of time delay of each action in figure.11.We compared the energy consumption of each hop in different actions, too. The result is shown in Figure 12. Both Fig.11 and Fig.12 show that ASDP is more sensitive and flexible with action's changes.

6 Related Work

In WSNs, the simplest form of service discovery is global flooding, which does not scale well. Many organizations have designed and developed service discovery protocols[16]. These still have not been mature enough to be used by industry for mobile ad hoc and sensor networks environment. However, many research works on Service discovery in MANET and WSNs have been conducted over the past few years. In [17], Dipanjan Chakraborty et al. introduced group based service discovery protocol which considered little mobility and energy efficiency. Yu Yang et al consider multicast-based SDPs may be more suitable for MANETs. They proposed HESED in [22]. Katsigiannis, C.O. et al. also proposed an architecture for reliable service discovery and delivery in MANETs based on power management employing SLP extensions. [18] Jui Chi Liang, Mobile Service Discovery Protocol (MSDP)for Mobile Ad-Hoc Networks .Marin-Perianu, proposed an Energy-Efficient Cluster-Based Service Discovery in WSNs via topology control scheme in [2][10]. Sethom. K. and Afifi. H also introduced a simple service discovery architecture for sensor networks[8].

Ant Colony Optimaization approach is proposed by Marco Dorigo, and has been used to resolve routing, localization and object tracing problems in mobile ad hoc and sensor networks. L.Zhang et al. proposed an ant colony based multicasting protocol in MANET to minimize the route overhead in [12]. In [15], Tiago Camilo proposed two kinds of energy-efficient ant colony based routing algorithms for WSNs, and compared the performance of these algorithms. In [20][21], Zheng X.Q and Liu L.G tried to use an ant colony based algorithm to design QoS-Aware Routing Algorithm for MANETs.Ricky Robinson et.al hold the same view with us and considered that ant colony and WSNs have many common characteristics. They were the first to propose a scheme to use ant colony approach to execute service discovery[6]. However, they just tried to use ant colony in the Service Advertisements and did not carry out a detailed evaluation.

7 Conclusions

WSNs have already opened a wide perspective for future applications especially in ubiquitous computing. Thus we identified service discovery as one of the major design components. Inspired by ant colony, we described a novel scalable Action-based Service Discovery Protocol (ASDP) using ant colony algorithm in WSNs in this paper. ADSP can abstract the semantics information from the data generation via the nodes or user's operation via the sinks and map them in to six different action sets. Then it adjusts the related parameters to satisfy with service and transmission requirements from different kinds of actions. We evaluated it against other approaches to identify its merits and limitations. The simulation results show that ASDP can maximize the network utilization. The farther experiments indicate it scales to large number of nodes. In the future, we plan to further improve the performance of ASDP under different movement, node failure or different traffic scenarios and try to summarize approach to adjust the parameters.

Acknowledgements

We would like to thank the NSFC for their financial support of the grant project (No. 60473001) Hongwei Huo would like to thank Beijing Jiaotong University for their financial support of the Ph.D Student Innovation Fund Program (No.48022).

References

1. Zhu, F., Mutka, M.W., Ni, L.M.: A Private, Secure, and User-Centric Information Exposure Model for Service Discovery Protocols. IEEE Transactions on Mobile Computing 5(4), 418–429 (2006)
2. Marin-Perianu, R., Scholten, H., Havinga, P., Hartel, P.: Energy-Efficient Cluster-Based Service Discovery in Wireless Sensor Networks. In: LCN 2006. Proceedings of 31st IEEE Conference on Local Computer Networks, pp. 931–938 (November 2006)
3. Katsigiannis, C.O., Kateros, D.A., Koutsoloukas, E.A., Tselikas, N.-L.D., Venieris, I.S.: Architecture for Reliable Service Discovery and Delivery in MANETs based on Power Management Employing SLP Extensions. IEEE Wireless Communications 13(5), 90–95 (2006)
4. Tyan, J., Mahmoud, Q.H.: A Comprehensive Service Discovery Solution for Mobile Ad Hoc Networks. Mobile Networks and Applications 10(4), 423–434 (2005)
5. Liljana, G., Ramjee, P.: Ad Hoc Networking Towards Seamless Communications, 1st edn. Ch. 6, pp. 143–172. Springer, Netherland (2006)
6. Robinson, R., Indulska, J.: A Complex Systems Approach to Service Discovery. In: Galindo, F., Takizawa, M., Traunmüller, R. (eds.) DEXA 2004. LNCS, vol. 3180, pp. 657–661. Springer, Heidelberg (2004)

7. Zhang, Y., Cao, J., Chan, A.T.S., Chan, K.C.C.: Sensors and Wireless Sensor Networks for Pervasive Computing Applications. Journal of Ubiqutous Computing and Inteligence 1(1), 17–34 (2007)
8. Sethom, K., Afifi, H.: A New Service Discovery Architecture for Sensor Networks. In: WTS 2005. Proceedings of Wireless Telecommunications Symposium, 2005, pp. 190–196 (April 2005)
9. Xinlian, Z., Mi, W.: Service Discovery Protocol in Wireless Sensor Networks. In: SKG 2006. Proceedings of the Second International Conference on Semantics, Knowledge, and Grid, pp. 101–102 (November 2006)
10. Marin-Perianu, R.S., Scholten, J., Havinga, P.J.M., Hartel, P.H.: Cluster-based Service Discovery for Heterogeneous Wireless Sensor Networks, Technical Report: TR-CTIT-07-05, University of Twente, Netherlands (May 2007)
11. Kamra, A., Misra, V., Feldman, J., Rubenstein, D.: Growth Codes: Maximizing Sensor Network Data Persistence. In: Sigcomm 2006. Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, pp. 255–266 (September 2006)
12. Zhang, L., Shen, D., Shan, X., Li, V.O.K.: An Ant-based Multicasting Procotol in Mobile Ad-Hoc Network. International Journal of Computational Intelligence and Applications 5(2), 185–199 (2005)
13. Marin-Perianu, M., Hofmeijer, T.J., Havinga, P.J.M.: Assisting Business Processes through Wireless Sensor Networks. In: ICT 2006. Proceedings of 13th International Conference on Telecommunications, pp. 6–10 (May 2006)
14. Dorigo, M., Stutzle, T.: Ant Colony Optimaization. Ch. 4, pp. 215–238. The MIT press, Cambridge (2004)
15. Camilo, T., Carreto, C., Silva, J.S., Boavida, F.: An Energy-Efficient Ant-Based Routing Algorithm for Wireless Sensor Networks. In: Dorigo, M., Gambardella, L.M., Birattari, M., Martinoli, A., Poli, R., Stützle, T. (eds.) ANTS 2006. LNCS, vol. 4150, Springer, Heidelberg (2006)
16. Zhu, F., Mutka, M.W., Ni, L.M.: Service Discovery in Pervasive Computing Environments. IEEE Pervasive Computing, 97–112 (December 2005)
17. Chakraborty, D., Joshi, A., Yesha, Y., Finin, T.: Toward Distributed Service Discovery in Pervasive Computing Environments. IEEE Transaction on Mobile Computing 5(2), 97–112 (2006)
18. Liang, J.C., Chen, J.C., Zhang, T.: Mobile Service Discovery Protocol (MSDP) for Mobile Ad-Hoc Networks. In: ISADS 07. Proceedings of 8th International Symposium on Autonomous Decentralized Systems, pp. 352–362 (April 2007)
19. Hussein, O., Saadawi, T.: Ant routing algorithm for mobile ad-hoc networks (ARAMA). In: IPCCC.2003. Proceedings of the 2003 IEEE International Performance, Computing, and Communications Conference, pp. 281–290 (April 2003)
20. Liu, L., Feng, G.: A Novel Ant Colony Based QoS-Aware Routing Algorithm for MANETs. In: Wang, L., Chen, K., Ong, Y.S. (eds.) ICNC 2005. LNCS, vol. 3612, pp. 457–466. Springer, Heidelberg (2005)
21. Xiangquan, Z., Lijia, G., Wei, G., Renting, L.: A cross-layer design and ant-colony optimization based load-balancing routing protocol for ad-hoc networks. Frontiers of Electrical and Electronic Engineering in China 2(2), 219–229 (2007)
22. Yang, Y., Hassanein, H., Mawji, A.: A New Approach to Service Discovery in Wireless Mobile Ad Hoc Networks. In: ICC 2006. Proceeding of 2006 IEEE International Conference on Communication, pp. 3838–3843 (June 2006)

Deadline Monotonic Policy over 802.11 Throughput and Average Service Time

Inès El Korbi and Leila Azouz Saidane

Ecole Nationale des Sciences de l'Informatique,
Université de la Manouba, Laboratoire Cristal, 2010 Tunisia
ines.korbi@gmail.com,
leila.saidane@ensi.rnu.tn

Abstract. In this paper, we propose a real time scheduling policy over 802.11 DCF protocol called Deadline Monotonic (DM). We evaluate the performance of this policy for a simple scenario where two stations with different deadlines contend for the channel. For this scenario a Markov chain based analytical model is proposed. From the mathematical model, we derive expressions of the saturation throughout and the average medium access delay called service time. Analytical results are validated by simulation results using the ns-2 network simulator.

1 Introduction

The IEEE 802.11 wireless LANs [2] become more and more reliable to support applications with Quality of Service (QoS) requirements. Indeed, the IEEE 802.11e standard [3] was recently proposed to offer service differentiation over 802.11.

In the absence of a coordination point, the IEEE 802.11 defines the Distributed Coordination Function (DCF) based on the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol. In the DCF protocol, a station shall ensure that the channel is idle when it attempts to transmit. Then it selects a random backoff in the contention window $[0, CW-1]$, where CW is the current window size and takes its values between the minimum and the maximum contention window sizes. If the channel is sensed busy, the station suspends its backoff until the channel becomes idle for a Distributed Inter Frame Space (DIFS) period after a successful transmission or an Extended Inter Frame Space (EIFS) period after a collision. When the backoff reaches 0, the packet is transmitted. A packet is dropped if it collides after maximum retransmission attempts.

The IEEE 802.11e standard proposes the Enhanced Distributed Channel Access (EDCA) as an extension for DCF. With EDCA, each station maintains four priorities called Access Categories (ACs). Each access category is characterized by a minimum and a maximum contention window sizes and an Arbitration Inter Frame Spacing (AIFS). Different analytical models have been proposed to evaluate the performance of 802.11e standard [9],[4], all inspired from Bianchi's model [1] that calculates saturation throughput of 802.11 DCF.

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 350–362, 2007.

Nevertheless, the granularity of service offered by 802.11e (4 priorities at most) can not satisfy the real time flows requirements (each flow is characterized by its own deadline). Therefore, we propose in this paper a new medium access mechanism based on the Deadline Monotonic (DM) policy [7] to schedule real time flows over 802.11.

To support the DM policy over 802.11, we introduce a distributed scheduling and a new medium access backoff policies. We then propose a mathematical model based on Markov chains analysis to evaluate the performance of DM for a simple scenario where two stations with different deadline constraints contend for the channel. This configuration will reflect the behavior of DM over 802.11 and the mathematical model can be extended for more complex scenarios. For the considered configuration, we evaluate for each station the saturation throughput and the average medium access delay called average service time. Analytical results will be validated against simulation results using the ns-2 simulator [8].

The rest of this paper is organized as follows. In section 2, we present the distributed scheduling and the new medium access backoff policy to support DM over 802.11. In section 3, we present our mathematical model based on Markov chain analysis. Section 4 and 5 provide analytical and simulation results of the throughput and the average service time respectively. Finally, we conclude the paper and present our future work in Section 6.

2 Supporting DM Policy over 802.11

Deadline Monotonic policy (DM) [7] is a real time scheduling policy that assigns static priorities to flow packets according to their deadlines; the packet with the small deadline being assigned the highest priority.

Indeed, when packets arrive to a station, they are sorted by increasing order of their deadlines such as the Head of Line (HOL) packet has the shortest deadline. The problem that occurs with the DCF is that all the stations share the same transmission medium and the HOL packets of all the stations will contend for the channel with the same priority even they have different deadlines.

The idea of introducing DM over 802.11 is to allow stations having packets with short deadlines to access the channel with higher priority than those having packets with long deadlines. Providing such a QoS requires a distributed scheduling and a new medium access policy.

2.1 Distributed Scheduling

To realize a distributed scheduling over 802.11, we introduce a broadcast priority mechanism similar to [5]. Indeed each station maintains a local scheduling table with entries for HOL packets of all other stations. Each entry in the scheduling table of node Si comprises two fields (Sj, Dj) where Sj is the source node MAC address (Address 2 field in DATA packet and RA field in the ACK packet) and Dj is the deadline of the HOL packet of Sj. To broadcast the HOL packet deadlines, we propose to use the DATA/ACK access mode. The deadline information requires two additional bytes to be encoded in DATA and ACK packets.

When a node Si transmits a DATA packet, it piggybacks the deadline of its HOL packet. The nodes hearing the DATA packet add an entry for Si in their local scheduling tables by filling the corresponding fields. The receiver of the DATA packet copies the priority of the HOL packet in the ACK before sending the ACK frame. All the stations that did not hear the DATA packet add an entry for Si using the information in the ACK packet.

In the following, we propose a new medium access policy, where the backoff value is inferred from the packet deadline.

2.2 DM Medium Access Backoff Policy

Let's consider two stations $S1$ and $S2$ transmitting two flows with the same deadline $D1$ ($D1$ is expressed as a number of 802.11 slots) such as $D1 < D2$, and generate two packets at time instants $t1$ and $t2$. If $S2$ had the same deadline as $S1$, its packet would have been generated at time $t2'$ such as $t2' = t2 + D21$, where $D21 = (D2 - D1)$. At that time $S1$ and $S2$ would have the same priority and transmit their packets according to the 802.11 protocol. Hence, when $S2$ has a packet to transmit, it selects a 802.11 backoff, but suspends this backoff during $D21$ idle slots. The $D21$ slots elapsed, 802.11 backoff can therefore be decremented.

Thus to support DM over 802.11, each station uses a new backoff policy where the backoff is given by:

- The random backoff selected in $[0, CW - 1]$, according to 802.11 DCF, referred as the BAsic Backoff (BAB).
- The DM Shifting Backoff ($DMSB$): corresponds to the additional backoff slots that a station with low priority adds to its BAB to have the same priority as the station with the highest priority.

Whenever a station Si sends an ACK or hears an ACK on the channel its $DMSB$ is reevaluated as follows:

$$DMSB(Si) = Deadline(HOL\,(Si)) - DT_{\min}(Si) \tag{1}$$

Where $DT_{\min}(Si)$ is the minimum of the HOL packet deadlines present in Si scheduling table and $Deadline(HOL\,(Si))$ is the HOL packet deadline of node Si.

Hence, when Si has to transmit its HOL packet with a deadline D_i, it selects a BAB in the contention window $[0, CW_{\min} - 1]$ and computes the WHole Backoff (WHB) value as follows:

$$WHB(Si) = DMSB(Si) + BAB(Si) \tag{2}$$

The station Si decrements its WHB whenever it senses an idle slot. Each time, the channel is sensed busy, Si reinitializes its $DMSB$ as in equation (2). Indeed, if a successful transmission is heard, all the stations reevaluate their

$DMSB$ when hearing a correct ACK and add the new $DMSB$ value to their current BAB. Whereas, if a collision is heard, all the stations will reinitialize their $DMSB$ and add it to their current BAB to allow colliding stations transmitting with the same priority as for their first transmission attempt. Si transmits when its WHB reaches 0.

If the transmission fails, Si doubles its contention window size, selects a BAB in the new window CW and repeats the above procedure until the packet is successfully transmitted or dropped after maximum retransmission attempts.

3 Mathematical Model of the DM Policy over 802.11

In this section, we propose a mathematical model to evaluate the performance of the DM policy using Markov chains analysis [1]. We consider the following assumptions:

Assumption1: The system under study comprises two stations $S1$ and $S2$, such as Si transmits a flow Fi having a deadline Di. We have $D1 < D2$ and $D21 = (D2 - D1)$.

Assumption2: We operate in saturation conditions: each station has always packets to transmit [1].

Assumption 3: A station selects a BAB in a contention window $[0, W - 1]$. We consider that each station selects a 802.11 backoff in the same contention window of size W independently of the transmission attempt. This is a simplifying assumption to limit the complexity of the mathematical model.

Assumption4: We are in stationary conditions, i.e. the two stations have already sent at least one packet to each other.

Each station Si will be modeled by a Markov Chain representing the whole backoff (WHB) process.

3.1 Markov Chain Modeling Station $S1$

Figure 1 represents the Markov Chain modeling station $S1$. The states of this Markov chain are described by the following quadruplet $(R, i, i - j, -D21)$ where:

- R : takes two values $\sim S2$ and $S2$. When $R = \sim S2$, $S2$ don't contend for the channel and decrements its $DMSB$ during $D21$ slots. When $R = S2$, $S2$ contends for the channel with $S1$.
- i : the value of the BAB selected by $S1$ in $[0, W - 1]$.
- $(i - j)$: corresponds to the current backoff of station $S1$.
- $(-D21)$: We choose the negative notation for $S1$ to express the fact that $S2$ has a positive $DMSB$ equals to $D21$ and $DMSB\,(S1) = 0$.

Initially $S1$ selects a random BAB and is in one of the states $(\sim S2, i, i, -D21)\,, i = 0..W-1$. During $(D21 - 1)$ slots, $S1$ decrements its backoff with the probability 1 and moves to one of the states $(\sim S2, i, i - j, -D21)$,

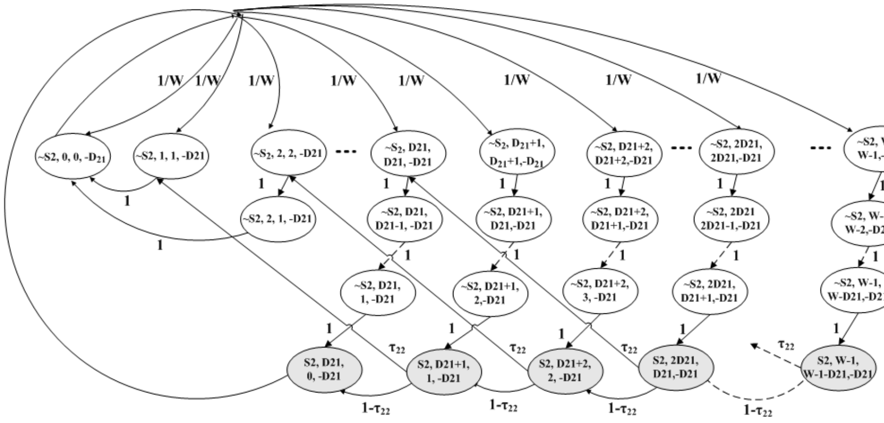


Fig. 1. Markov Chain modeling station $S1$

$i = 0..(W-1)$, $j = \min(\max(0, i-1), D21-1)$. Indeed during these slots, $S2$ is decrementing its $DMSB$ and wouldn't contend for the channel. When $S1$ decrements its $D21^{th}$ slot it knows that henceforth, $S2$ can contend for the channel (the $D21$ slots were elapsed). Hence, $S1$ moves to one of the states $(S2, i, i-D21, -D21)$, $i = D21..W-1$. If the BAB initially selected by $S1$ is smaller than $D21$, then $S1$ transmits when its backoff reaches 0. If $S2$ transmits before $S1$ backoff reaches 0, the next packet of $S2$ will decrement another $DMSB$ and $S1$ will see the channel free again for $D21$ slots.

3.2 Markov Chain Modeling Station $S2$

Figure 2 represents the Markov Chain modeling station $S2$. Each state of $S2$ Markov chain is represented by the quadruplet $(i, k, D21-j, D21)$ where:

- i : refers to the BAB value selected by $S2$ in $[0, W-1]$.
- k : refers to the current BAB value.
- $D21-j$: refers to the current $DMSB$ of $S2$, $j \in [0, D21]$.
- $D21$: corresponds to $(D2-D1)$.

When $S2$ selects a BAB, its $DMSB$ equals $D21$ and is in one of states $(i, i, D21, D21)$, $i = 0..W-1$. If $S2$ observes the channel idle during $D21$ slots, it moves to one of the states $(i, i, 0, D21)$, $i = 1..W-1$, where it ends its shifting backoff and begins decrementing its basic backoff. If $S1$ transmits, $S2$ reinitializes its shifting backoff and moves again to one of the states $(i, i, D21, D21)$, $i = 1..W-1$.

3.3 Blocking Probabilities in the Markov Chains

We notice from figure 1 that when $S1$ is in one of the states $(\sim S2, i, i-j, -D21)$, $i = 0..W-1$, $j = \min(\max(0, i-1), D21-1)$, it decrements its backoff with

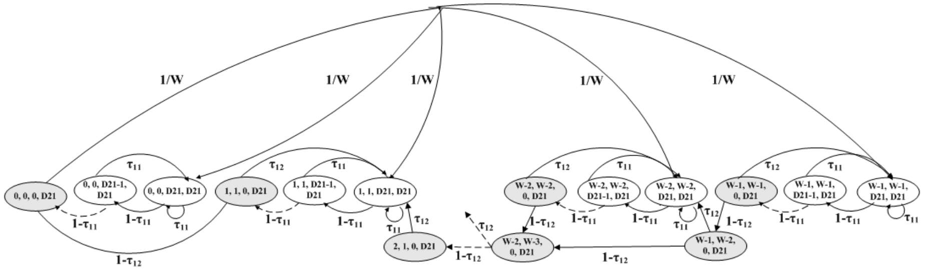


Fig. 2. Markov Chain modeling station $S2$

the probability 1. It means that when $S1$ is in one of these states, it knows that $S2$ is decrementing its $DMSB$ and is in one of the states $(i, i, D21 - j, D21)$, $i = 0..W - 1$, $j = 0..(D21 - 1)$.

However, when $S1$ is in one of the states $(S2, i, i - D21, -D21)$, $i = D21..(W - 1)$, $S2$ has already decremented its $DMSB$ and can now contend for the channel by decrementing its basic backoff. In this case, $S2$ will be in one of the states $(i, i, 0, D21)_{i=0..W-1} \cup (i, i - 1, 0, D21)_{i=2..W-1}$. From the explanations above, each station Markov chain states can be divided in two groups:

- ξ_1 : the set of states of $S1$ for which $S2$ will not contend (white states in figure 1).
  $\xi_1 = \left\{(\sim S2, i, i - j, -D21)_{i=0..W-1, j=0..\min(\max(0,i-1),D21-1)}\right\}$.
- γ_1 : the set of states of station $S1$ for which $S2$ can contend and decrements its BAB (grey states in figure 1).
  $\gamma_1 = \left\{(\ S2, i, i - D21, -D21)_{i=D21..W-1}\right\}$.
- ξ_2 : the set of states of $S2$ where $S2$ does not contend for the channel (white states in figure 2).
  $\xi_2 = \left\{(i, i, D21 - j, D21)_{i=0..W-1, j=0..D21-1}\right\}$.
- γ_2 : the set of states of $S2$, where $S2$ contends for the channel (grey states in figure 2).
  $\gamma_2 = \left\{(i, i, 0, D21)_{i=0..W-1} \cup (i, i - 1, 0, D21)_{i=2..W-1}\right\}$.

Thus when $S1$ is in one of the states of ξ_1, $S2$ is obligatory in one of the states of ξ_2. Similarly, when $S1$ is in one of the states of γ_1, $S2$ is obligatory in one of the states of γ_2.Thus, the blocking probabilities τ_{11}, τ_{12} and τ_{22} described in figures 1 and 2 are given by:

$$\tau_{11} = \Pr\left[S1 \text{ transmits} / \xi_1\right] = \frac{\pi_1^{(\sim S2,0,0,-D21)}}{\sum\limits_{i=0}^{W-1}\left(\sum\limits_{j=0}^{\min(\max(0,i-1),D21-1)} \pi_1^{(\sim S2,i,i-j,-D21)}\right)} \qquad (3)$$

$$\tau_{12} = \Pr\left[S1 \text{ transmits} /\gamma_1\right] = \frac{\pi_1^{(S2,D21,0,-D21)}}{\sum\limits_{i=D21}^{W-1} \pi_1^{(S2,i,i-D21,-D21)}} \tag{4}$$

$$\tau_{22} = \Pr\left[S2 \text{ transmits} /\gamma_2\right] = \frac{\pi_2^{(i,0,0,D21)}}{\sum\limits_{i=0}^{W-1} \pi_2^{(i,i,0,D21)} + \sum\limits_{i=2}^{W-1} \pi_2^{(i,i-1,0,D21)}} \tag{5}$$

Where $\pi_1^{(R,i,i-j,-D21)}$ (respectively $\pi_2^{(i,k,i-j,D21)}$) is the probability of the state $(R, i, i-j, -D21)$ (respectively the probability of the state $(i, k, i-j, D21)$) in the stationary conditions and $\Pi_1 = \left\{\pi_1^{(R,i,i-j,-D21)}\right\}$ (respectively $\Pi_2 = \left\{\pi_2^{(i,k,i-j,D21)}\right\}$) is the probability vector of station $S1$ (respectively the probability vector of station $S2$).

The blocking probabilities described above allow deducing the transition state probabilities and having the transition probability matrix P_i, for each station Si. Then, we evaluate the state probabilities by solving the following system [6]:

$$\begin{cases} \Pi_i P_i = \Pi_i \\ \sum\limits_j \pi_i^j = 1 \end{cases} \tag{6}$$

3.4 Transition Probability Matrices

Transition probability matrix of $S1$: Let P_1 be the transition probability matrix of $S1$ and $P_1\{i, j\}$ is the probability to transit from state i to state j. The transitions probabilities of station $S1$ are:

$$\begin{aligned} &P_1\left\{(\sim S2, i, i-j, -D21), (\sim S2, i, i-(j+1), -D21)\right\} = 1, \\ &i = 2..W-1, j = 0..\min\left(\max\left(0, i-2\right), D21-2\right) \end{aligned} \tag{7}$$

$$P_1\left\{(\sim S2, i, i-D21+1, -D21), (S2, i, i-D21, -D21)\right\} = 1, i = D21..W-1 \tag{8}$$

$$P_1\left\{(\sim S2, i, 1, -D21),\ (\sim S2, 0, 0, -D21)\right\} = 1, i = 1..\min\left(W-1, D21-1\right) \tag{9}$$

$$\begin{aligned} &P_1\left\{(S2, i, i-D21, -D21), (S2, i-1, i-1-D21, -D21)\right\} = 1-\tau_{22}, \\ &i = D21+1..W-1 \end{aligned} \tag{10}$$

$$\begin{aligned} &P_1\left\{(S2, i, i-D21, -D21), (\sim S2, i-D21, i-D21, -D21)\right\} = \tau_{22}, \\ &i = D21+1..W-1 \end{aligned} \tag{11}$$

$$P_1\left\{(\sim S2, 0, 0, -D21), (\sim S2, i, i, -D21)\right\} = \frac{1}{W},\ i = 0..W-1 \tag{12}$$

If $(D21 < W)$then:

$$P_1\left\{(S2, D21, 0, -D21), (\sim S2, i, i, -D21)\right\} = \frac{1}{W},\ i = 0..W-1 \tag{13}$$

By replacing P_1 and Π_1 in (6) and solving the resulting system, we can express $\pi_1^{(R,i,i-j-D21)}$ as a function of τ_{22}, where τ_{22} is given by (5).

Transition probability matrix of *S*2: Let P_2 be the transition probability matrix of $S2$. The transitions probabilities of $S2$ are:

$$P_2\left\{(i,i,D21-j,D21),(i,i,D21-(j+1),D21)\right\}=1-\tau_{11},\\ i=0..W-1, j=0..(D21-1) \tag{14}$$

$$P_2\left\{(i,i,D21-j,D21),(i,i,D21,D21)\right\}=\tau_{11},\ i=0..W-1, j=0..(D21-1) \tag{15}$$

$$P_2\left\{(i,i,0,D21),(i,i-1,0,D21)\right\}=1-\tau_{12},\ i=2..W-1 \tag{16}$$

$$P_2\left\{(1,1,0,D21),(0,0,0,D21)\right\}=1-\tau_{12} \tag{17}$$

$$P_2\left\{(i,i,0,D21),(i,i,D21,D21)\right\}=\tau_{12}, i=1..W-1 \tag{18}$$

$$P_2\left\{(i,i-1,0,D21),(i-1,i-1,D21,D21)\right\}=\tau_{12},\ i=2..W-1 \tag{19}$$

$$P_2\left\{(i,i-1,0,D21),(i-1,i-2,0,D21)\right\}=1-\tau_{12},\ i=3..W-1 \tag{20}$$

$$P_2\left\{(0,0,0,D21),(i,i,D21,D21)\right\}=\frac{1}{W},\ i=0..W-1 \tag{21}$$

Replacing P_2 and Π_2 in (6) and solving the resulting system, we can express $\pi_2^{(i,k,D21-j,D21)}$ as a function of τ_{11} and τ_{12} given respectively by (3) and (4). Moreover, by replacing $\pi_1^{(R,i,i-j,D21)}$and $\pi_2^{(i,k,D21-j,D21)}$ by their values, in equations (3), (4) and (5), we obtain a system of non linear equations as follows:

$$\begin{cases}\tau_{11}=f\left(\tau_{22}\right)\\ \tau_{12}=f\left(\tau_{22}\right)\\ \tau_{22}=f\left(\tau_{11},\tau_{12}\right)\\ \textit{under the constraint:}\\ \tau_{11}>0,\tau_{12}>0,\tau_{22}>0,\tau_{11}<1,\tau_{12}<1,\tau_{22}<1\end{cases} \tag{22}$$

Solving the above system (22), allows deducing the expressions of τ_{11},τ_{12} and τ_{22}, and deriving the state probabilities of $S1$ and $S2$ Markov chains.

4 Throughput Analysis

In this section, we propose to evaluate B_i, the normalized throughput achieved by each station Si [1]. Hence, we define:

- $P_{i,s}$: the probability that station Si transmits a packet successfully.

$$P_{1,s}=\Pr\left[\ S1\text{ transmits successfully }/\xi_1\right]\Pr\left[\xi_1\right]\\ +\Pr\left[\ S1\text{ transmits successfully }/\gamma_1\right]\Pr\left[\gamma_1\right]=\tau_{11}\Pr\left[\xi_1\right]+\tau_{12}\left(1-\tau_{22}\right)\Pr\left[\gamma_1\right] \tag{23}$$

$$P_{2,s}=\Pr\left[\ S2\text{ transmits successfully }/\xi_2\right]\Pr\left[\xi_2\right]\\ +\Pr\left[\ S2\text{ transmits successfully }/\gamma_2\right]\Pr\left[\gamma_2\right]=\tau_{22}\left(1-\tau_{12}\right)\Pr\left[\gamma_2\right] \tag{24}$$

– P_{idle} : the probability that the channel is idle.

$$P_{idle} = (1 - \tau_{11}) \Pr[\xi_1] + (1 - \tau_{22})(1 - \tau_{12}) \Pr[\gamma_1] \tag{25}$$

Hence, the expression of the station Si throughput is given by:

$$B_i = \frac{P_{i,s} T_p}{P_{idle} T_e + T_s \sum_{i=1}^{2} P_{i,s} + \left(1 - P_{idle} - \sum_{i=1}^{2} P_{i,s}\right) T_c} \tag{26}$$

Where T_e denotes the duration of an empty slot, T_s and T_c denote respectively the duration of a successful transmission and a collision. $\left(1 - P_{idle} - \sum_{i=1}^{2} P_{i,s}\right)$ corresponds to the probability of collision. Finally T_p denotes the average time required to transmit the packet data payload. We have:

$$T_s = (T_{PHY} + T_{MAC} + T_p + T_D) + SIFS + (T_{PHY} + T_{ACK} + T_D) + DIFS \tag{27}$$

$$\begin{aligned} T_c &= (T_{PHY} + T_{MAC} + T_p + T_D) + T_{EIFS} \\ &= (T_{PHY} + T_{MAC} + T_p + T_D) + (SIFS + (T_{PHY} + T_{ACK} + T_D) + DIFS) \end{aligned} \tag{28}$$

Where T_{PHY}, T_{MAC} and T_{ACK} are the durations of the PHY header, the MAC header and the ACK packet. T_D is the time required to transmit the two bytes deadline information.

For numerical results, $S1$ and $S2$ transmit 512 bytes data packets using 802.11.b MAC and PHY layers'parameters with a data rate equal to 11Mbps. For simulation scenarios, the transmission range of each node covers 250m and the distance between $S1$ and $S2$ is 5m. In figure 3, we represent the saturation

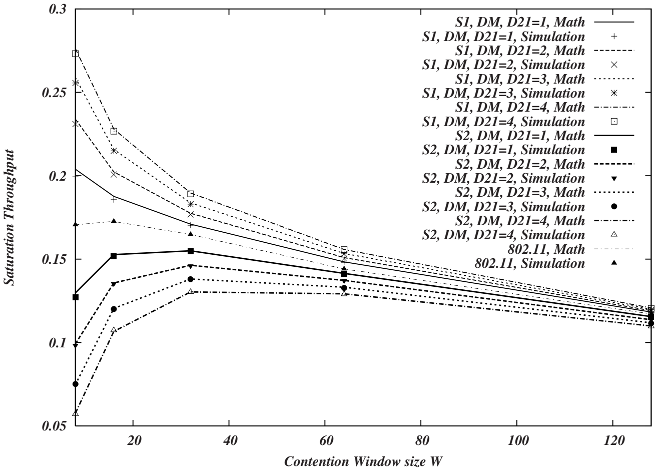


Fig. 3. Normalized throughput vs the contention window size

throughput of $S1$ and $S2$ as a function of the contention window size W, and for different values of $D21$. We validate analytical results by simulation using the ns-2 simulator [8]. Figure 3 shows that the throughput of station $S1$ is always greater than the one of $S2$. Indeed as $D21$ increases, $S1$ throughput increases whereas $S2$ throughput decreases. Moreover as the contention window size increases the difference between stations throughputs is reduced. This is due to the fact that the shifting backoff becomes negligible compared to the contention window size. Finally, we notice that $S1$ obtains better throughput with DM than with 802.11, but the opposite scenario happens for $S2$.

5 Average Service Time Computation

In this section, we evaluate the average MAC layer service time of $S1$ and $S2$ using DM policy. The service time is the time interval from the time instant that a packet becomes at the head of the queue and starts to contend for transmission to the time instant that either the packet is acknowledged for a successful transmission or dropped. We propose to evaluate the Z-Transform of the MAC layer service time [10] to derive an expression of the average service time. The average service time depends on the duration of an idle slot T_e, the duration of a successful transmission T_s and the duration of a collision T_c. As T_e is the smallest duration event, the duration of all events will be given by $\lceil \frac{T_{event}}{T_e} \rceil$.

5.1 Z-Transform of the MAC Layer Service Time

Z-transform of $S1$ service time: To evaluate the Z-transform of station $S1$ service time $TS_1(Z)$, we define:

$H1_{(R,i,i-j,-D21)}(Z)$: The Z-transform of the time already elapsed from the instant $S1$ selects a basic backoff in $[0, W-1]$ (i.e. being in one of the states $(\sim S2, i, i, -D21)$) to the time it is found in the state $(R, i, i-j, -D21)$. Thus:

$$H1_{(\sim S2,W-1,W-1-j,-D21)}(Z) = \frac{1}{W} Z^j, j = 0..D21 \tag{29}$$

$$H1_{(\sim S2,i,i,-D21)}(Z) = \frac{1}{W} Z, i = (W-D21)..(W-2) \tag{30}$$

$$\begin{aligned} &H1_{(\sim S2,i,i,-D21)}(Z) = \tau_{22} Z^{\lceil \frac{T_s}{T_e} \rceil} H1_{(S2,i+D21,i,-D21)}(Z) + \tfrac{1}{W}, \\ &i = 1..\max(0, (W-1-D21)) \end{aligned} \tag{31}$$

$$\begin{aligned} &H1_{(\sim S2,i,i-j,-D21)}(Z) = Z^j H1_{(\sim S2,i,i,-D21)}(Z), \\ &i = 1..W-2, j = 1..\min(i-1, D21-1) \end{aligned} \tag{32}$$

$$\begin{aligned} &H1_{(S2,i,i-D21,-D21)}(Z) = Z^{D21} H1_{(\sim S2,i,i,-D21)}(Z) \\ &+ (1-\tau_{22}) Z H1_{(S2,i+1,i+1-D21,-D21)}(Z), i = D21..W-2 \end{aligned} \tag{33}$$

$$H1_{(\sim S2,0,0,-D21)}(Z) = \frac{1}{W} + \sum_{i=1}^{\min(W-1,D21-1)} H1_{(\sim S2,i,1,-D21)}(Z) \tag{34}$$

If $S1$ transmission state is $(\sim S2, 0, 0, -D21)$, the transmission will be successful since $S2$ was decrementing its shifting backoff. Whereas when the station $S1$ transmission state is $(S2, D21, 0, -D21)$, the transmission occurs successfully only if $S2$ doesn't transmit with the probability $(1 - \tau_{22})$. Otherwise $S1$ selects another backoff and tries another transmission. After m retransmissions, if the packet is not acknowledged, it will be dropped. So :

$$\begin{aligned} TS_1(Z) &= Z^{\lceil \frac{Ts}{Te} \rceil} \left(H1_{(\sim S2,0,0,-D21)}(Z) \right. \\ &\left. + (1-\tau_{22}) H1_{(S2,D21,0,-D21)}(Z) \right) \sum_{i=0}^{m} \left(\tau_{22} Z^{\lceil \frac{Tc}{Te} \rceil} H1_{(\ S2,D21,0,-D21)}(Z) \right)^{i} \\ &+ \left(\tau_{22} Z^{\lceil \frac{Tc}{Te} \rceil} H1_{(\ S2,D21,0,-D21)}(Z) \right)^{m+1} \end{aligned} \tag{35}$$

Z-transform of $S2$ service time: In the same way, we define $TS_2(Z)$, the Z-transform of station $S2$ service time and:

$H2_{(i,k,D21-j,-D21)}(Z)$: The Z-transform of the time already elapsed from the instant $S2$ selects a basic backoff in $[0, W-1]$ (i.e. being in one of the states $(i, i, D21, D21)$) to the time it is found in the state $(i, k, D21-j, -D21)$.

$$H2_{(i,i,D21,D21)}(Z) = \frac{1}{W}, i = 0 \text{ and } i = W-1 \tag{36}$$

$$H2_{(i,i,D21,D21)}(Z) = \left(\frac{1}{W} + \tau_{12} Z^{\lceil \frac{Ts}{Te} \rceil} H1_{(i+1,i,0,D21)}(Z) \right), i = 1..\ W-2 \tag{37}$$

To compute $H2_{(i,i,D21-j,D21)}(Z)$, we define $T_{dec}^{j}(Z)$, such as:

$$T_{dec}^{0}(Z) = 1 \tag{38}$$

$$T_{dec}^{j}(Z) = \frac{(1-\tau_{11}) Z}{1 - \tau_{11} Z^{\lceil \frac{Ts}{Te} \rceil} T_{dec}^{j-1}(Z)}, j = 1..D21 \tag{39}$$

So:

$$\begin{aligned} &H2_{(i,i,D21-j,D21)}(Z) = H2_{(i,i,D21-j+1,D21)}(Z)\, T_{dec}^{j}(Z), \\ &i = 0..W-1, j = 1..D21 \end{aligned} \tag{40}$$

$$H2_{(i,i-1,0,D21)}(Z) = \frac{(1-\tau_{12}) Z H2_{(i,i,0,D21)}(Z)}{1 - \tau_{12} Z^{\lceil \frac{Ts}{Te} \rceil} T_{dec}^{D21}(Z)}, i = 2..W-1 \tag{41}$$

We also have :

$$\begin{aligned} &H2_{(i,i-1,0,D21)}(Z) = (1-\tau_{12}) Z \left(H2_{(i+1,i,0,D21)}(Z) + H2_{(i,i,0,D21)}(Z) \right), \\ &i = 2..W-2 \end{aligned} \tag{42}$$

$$H2_{(W-1,W-2,0,D21)}(Z) = (1-\tau_{12}) Z H2_{(W-1,W-1,0,D21)}(Z) \tag{43}$$

According to figure 1 and equation (36):

$$H2_{(0,0,0,D21)}(Z) = H2_{(0,0,0,D21)}(Z) + \frac{(1-\tau_{12})\, Z H2_{(1,1,0,D21)}(Z)}{1-\tau_{12} Z^{\lceil \frac{T_s}{T_e} \rceil} T_{dec}^{D21}(Z)} \tag{44}$$

Therefore, the Z transform service time of station $S2$ is:

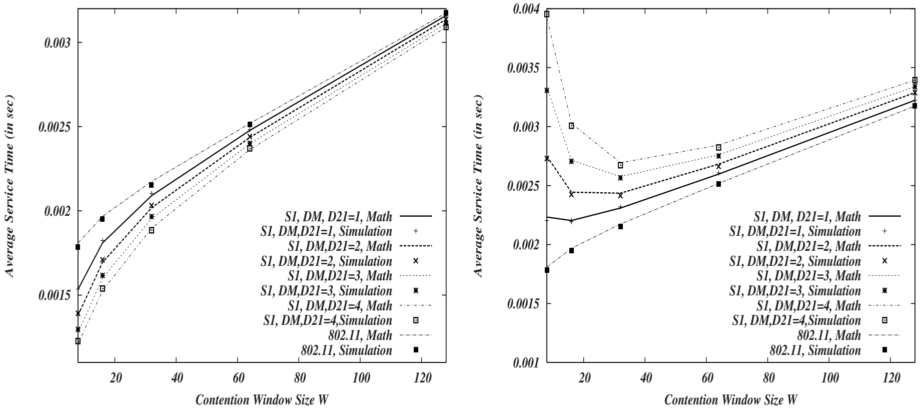
$$\begin{aligned} TS_2(Z) &= (1-\tau_{12})\, Z^{\lceil \frac{T_s}{T_e} \rceil} H2_{(0,0,0,D21)}(Z) \sum_{i=0}^{m} \left(\tau_{12} Z^{\lceil \frac{T_c}{T_e} \rceil} H2_{(0,0,0,D21)}(Z) \right)^{i} \\ &+ \left(\tau_{12} Z^{\lceil \frac{T_c}{T_e} \rceil} H2_{(0,0,0,D21)}(Z) \right)^{m+1} \end{aligned} \tag{45}$$

5.2 Average Service Time

From equations (35) and (45), we derive the average service time of both stations $S1$ and $S2$. The average service time of station Si is denoted by $\overline{X_i}$ and given by:

$$\overline{X_i} = TS_i^{(1)}(1) \tag{46}$$

Where $TS_i^{(1)}(Z)$, is the derivate of the service time Z-transform of station Si [6]. Figures 4(a) and 4(b) represent the average service time of both stations $S1$ and $S2$ as a function of the contention window size W for different values of $D21$. We also compared the average service time of $S1$ and $S2$ to the one obtained with 802.11. Figures 4(a) and 4(b) show that all the curves of the average service time of $S1$ are below the of one of 802.11 and the curves of the average service time of $S2$ are above the one of 802.11. We can therefore conclude that compared to 802.11, the DM policy offers better delay guarantees for the flow with the small deadline. Moreover the average service time of the priority flow becomes smaller as the difference between the deadlines of both flows increases.



(a) ***StationS1*** **(b)** ***Station S2***

Fig. 4. Average Service Time

6 Conclusion

In this paper we proposed to support the DM policy over 802.11 protocol. Therefore, we used a distributed scheduling algorithm and introduced a new medium access backoff policy. Then we proposed a mathematical model to evaluate the performance of the DM policy for a scenario where two stations with different deadlines contend for the channel. Analytical and simulation results show that the station with the shortest deadline has the highest saturation throughput and the lowest average medium access delay. Hence, we can conclude that DM performs service differentiation over 802.11 and offers better guarantees in terms of throughput and average service time for the flow having the small deadline. In future works, we intend to generalize the deadline monotonic analytical model to n contending stations transmitting flows with different deadlines.

References

1. Bianchi, G.: Performance Analysis of the IEEE 802.11 Distributed Coordination Function. IEEE J-SAC 18(3), 535–547 (2000)
2. IEEE 802.11 WG, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specification, IEEE (1999)
3. IEEE 802.11 WG, Draft Supplement to Part 11: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS), IEEE 802.11e/D13.0 (January 2005)
4. Engelstad, P.E., Østerbø, O.N.: Non-Saturation and Saturation Analysis of IEEE 802.11e EDCA with Starvation Prediction. In: MSWiM 2005, Canada (October 2005)
5. Kanodia, V., Li, C.: Distribted Priority Scheduling and Medium Access in Ad-hoc Networks. ACM Wireless Networks 8 (November 2002)
6. Kleinrock, L.: Queuing Systems: Theory, vol. 1. John Wiley, Chichester (1975)
7. Leung, J.Y.T., Whitehead, J.: On the Complexity of Fixed-Priority Scheduling of Periodic, Real-Time Tasks, Performance Evaluation (December 1982)
8. The network simulator - ns-2, http://www.isi.edu/nsnam/ns/
9. Xiao, Y.: Performance analysis of IEEE 802.11e EDCF under saturation conditions. In: Proceedings of ICC, Paris, France (June 2004)
10. Zhai, H., Kwon, Y., Fang, Y.: Performance Analysis of IEEE 802.11 MAC protocol in wireless LANs, Wireless Computer and Mobile Computing (2004)

A Parallel Link State Routing Protocol for Mobile Ad-Hoc Networks

Dong Yang*, Hongke Zhang, Hongchao Wang, Bo Wang, and Shuigen Yang

Beijing Jiaotong University, 100044, Beijing, China
youngmanyd@163.com, zhk@telecom.njtu.edu.cn,
{05111041,04111032,04111026}@bjtu.edu.cn

Abstract. With the network size growing, routing protocols of mobile ad-hoc networks (MANET) face many challenges. The most critical issue is the lack of bandwidth and computation capability. This paper describes the Parallel Link State Routing protocol (PLSR) that is suitable for MANET. The key concept used in the protocol is that of graph decomposition. This technique substantially raises the rate of routing updating; meanwhile it reduces the message overload as compared with a classical flooding mechanism and some other link state routing protocols. In PLSR, the network is a two-levels topology, and many problems caused by "flat" multi-hoppings will disappear. Thus, the second optimization is achieved by sub-network that reduces the control messages. As a third optimization, the routing computation will be in several sub-networks simultaneity. Hence, comparing with current link state algorithms, the routing table of PLSR will be regenerated quickly as the network topology changes.

Keywords: ad-hoc network, routing, link state.

1 Introduction

MANET is a collection of mobile nodes that can establish instant communication of civilian and military applications. As the size of MANET grows, the performance tends to decrease. One of the critical issues in ad-hoc networking is the lack of bandwidth and computation capability. So how to reduce the traffic overload and the pressure of computation is a very important design in MANET. Many routing protocols have been proposed for efficient ad-hoc routing. Existing ad-hoc routing protocols can be mainly classified into three different types [1]: (1) Table driven routing protocols, such as OLSR (optimized link state routing) [2], and TBRPF (topology-based reverse path forwarding) [3] based on link state; DBF (distributed Bellman-Ford) [4], and DSDV (destination sequenced distance vector) [5] based on distance vector. (2) Source initiated on-demand routing protocols, such as AODV (ad-hoc on-Demand distance vector routing) [6], TORA (temporary ordered routing algorithm) [7], DSR (dynamic source routing) [8], and ABR (associativity based

* This work was supported by the National Basic Research Program of China (No.2007CB-307101, No.2007CB301106), National Natural Science Foundation of China (No.60473001), and Key Project of Chinese Ministry of Education (No.706005).

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 363–374, 2007.

routing protocol) [9]. (3) Hybrid routing protocols, such as ZRP (zone routing protocol) [10]. Table driven routing protocol is proactive, and it requires each node to maintain one or more tables to store routing information. They respond to changes in network topology by propagating updates throughout the network in order to maintain a consistent network view. The areas in which they differ are the number of necessary routing-related tables and the methods by which changes in network structure are broadcasted. Source initiated on-demand routing protocol creates routings only when desired by the source node. When a node requires a routing to a destination, it initiates a routing discovery process within the network. This process is completed once a routing is found or all possible routing permutations have been examined. After a routing has been established, it is maintained by a routing maintenance procedure until either the destination becomes inaccessible or the routing is no longer desired. Hybrid routing protocol uses proactive protocol in local zone and reactive protocol between zones.

One kind of table driven routing protocols is the link state routing protocol. Classic link state routing protocol has been the dominating internal gateway Protocol for many years. With the development of ad-hoc network, the need for light, efficient and robust routing makes it a good candidate in this constrained environment. OLSR is one kind of link state routing protocol for MANET, and is considered as one of the most prominent ad-hoc networking solutions. TBRPF is also a popular link state routing protocol.

In this paper we present Parallel Link State Routing (PLSR), a new link state routing protocol for MANET, which is a parallel shortest path algorithm based on graph decomposition. Comparing with current MANET link state routing protocols such as OLSR, TBRPF, PLSR uses a different method to decrease overload and increase computation capability.

The remainder of this paper is organized as follows. Section 2 gives an overview of some existing link state routing algorithms for MANET. Section 3 presents the basis of parallel shortest path algorithm. Section 4 describes the routing algorithm of PLSR in detail and discusses its implementation. Subsequently, section 5 presents some simulation results to show how to optimize PLSR and compares it with existing routing algorithms. Finally, section 6 concludes the paper.

2 Current Link State Routing Protocol Used for MANET

OLSR protocol is one kind of table-driven protocol, and it is an improvement over the pure link state protocol. As a link state protocol, nodes of OLSR collect the whole topology information to compute routing between other nodes. To adapt to the low bandwidth and computation capability requirement of ad-hoc networks, OLSR makes some optimizations to reduce the overload required in establishing routings. Firstly, it minimizes flooding by the use of the MPR (Multipoint Relay) concept. Every node selects a set of nodes among its one-hop neighbors as MPRs. The node that selects this set of MPRs is called the MPR Selector. The function of the MPR is to forward the flooded message from its MPR Selector. When a node receives a control message, it only forwards the control message if it is an MPR of the sending node. The other non-MPR nodes will only receive but not forward the control messages, thereby retransmission and network traffic are decreased. MPRs of node *N* must satisfy following requests: the two-hops neighbors of *N* must have links to these MPRs. Secondly, they reduce the size of the control message by including only its MPR Selectors in its control message. Furthermore, OLSR does not generate any additional control messages in response to network topology change, other than the periodic control messages described above. These optimizations of OLSR make it suitable for large and dense network.

TBRPF is another kind of proactive, link-state routing protocol designed for MANET. TBRPF provides hop-by-hop routing along shortest paths to each destination. Unlike classic link state routing protocol, the nodes running TBRPF compute a source tree based on only partial topology information to minimize overload, using a modification of Dijkstra's algorithm. TBRPF uses a combination of periodic and differential updates to keep all neighbors informed of the reported part of its source tree. At the same time, TBRPF node has the option to report additional topology information to provide improved robustness in highly mobile networks. TBRPF performs neighbor discovery using differential HELLO messages that report only changes in the status of neighbors. This results in HELLO messages that are much fewer than those of other link state routing protocols such as OSPF.

Because of the restriction of lacking bandwidth, current MANET link state routing protocols use different technologies to reduce the overload based on classic link state routing protocol. In OLSR only MPR nodes forward topology information, and in TBRP each node reports only part of its source tree to neighbors to decrease overload. PLSR divides network into some sub-graphs to reduce the overload.

3 Parallel Shortest Path Algorithm Based on Graph Decomposition

Like many link state routing protocols, PLSR uses shortest path algorithm to compute routing. There are many parallel shortest path algorithms, such as [11] [12]. In PLSR, The parallel algorithm is an optimization of a hierarchical network decomposition algorithm [13] tailored to the requirements of MANET. The core of this article is dividing up the network into pieces in parallel to yield approximate shortest paths in large-scale networks.

We will describe the algorithm considering the following simple model. *G=(V, A)* is a graph, and every non-boundary node has exactly 6 neighbors. More precisely, suppose *G* is topologically equivalent to a mesh with *N* nodes (figure 1). We will aggregate nodes by forming a partition of the nodes of the network into *M* classes (In figure 1 there are 7 classes *A*, *B*, *C*, *D*, *E*, *F*, *G*) called macro-nodes. Moreover, we aggregate in such a way that the macro-network of *M* macro-nodes is a mesh with *M* nodes, and that every macro-node itself is a mesh with (*N/M*) nodes. A macro-arc is present between two macro-nodes if and only if there is an arc connecting two nodes in their respective aggregate classes. Define the arc lengths in the macro-network to be the shortest of the lengths of all micro-arcs connecting two macro-nodes.

For this simple model, we could approximately compute the shortest path by finding all shortest paths in the macro-network firstly, and then computing shortest paths within each macro-node. Finally we combine these two path sets to generate shortest paths between all nodes. The shortest paths based on this method are not the optimal, even if all the paths in a macro-network are solved optimally. Because of the computations in multiple macro-networks simultaneously, it is a parallel solution for shortest path problem. This is the graph decomposition algorithm to compute shortest path that is used in PLSR.

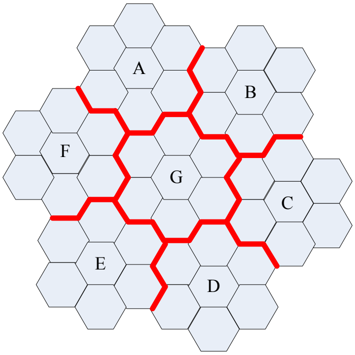


Fig. 1. Simple Model for Graph Decomposition

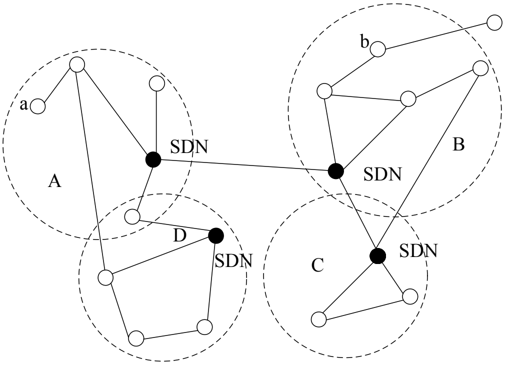


Fig. 2. Simple Model of PLSR

4 PLSR

The key concept of PLSR is graph decomposition. Figure 2 is a simple model of PLSR. The whole network is decomposed four sub-graphs *A*, *B*, *C*, *D*. (Section 4.4 will give a method of graph decomposition.) Every sub-graph has a delegation node called SDN (sub-graph delegation node). In PLSR node only maintain the topology information of the sub-graph that the node belongs to, and the nodes will get the routing to other nodes in same sub-graph using classic link state routing protocol. Meanwhile all SDNs maintain a topology of SDN set in the whole network topology, and they will get the routing between themselves. SDNs should maintain two routing tables, one is for its sub-graph and the other is used between SDNs. All the routing computation in sub-graph or between SDNs is parallel. After the routing computation, any two nodes in the same sub-graph will get a shortest path to communicate each other, and nodes in different sub-graph will get a sub-optimal shortest path that is connected by two or more SDNs.

There is a simple example based on figure 2 to illustrate the routing computation of PLSR. The node *a* wants to communicate with node *b*, and routing must be computed between node *a* and *b* first. In PLSR the whole network is divided into some sub-graphs. In figure 2 node *a* belongs to sub-figure *A*, and node *b* belongs to sub-figure *B*. The node *a* and *b* compute the shortest path to SDNs in their sub-graph. At the same time there are shortest path between SDNs in different sub-graphs. Finally, the routing between node *a* and *b* is comprised of three parts: from *a* to its SDN, between two SDNs, and from SDN of *b* to *b*. The routing is sub-optimal compared with classic link state routing protocol.

As link state routing protocol, OLSR and PLSR have some similarities. In this paper we will reference to some mature principles of OSLR for PLSR design.

From PSLR setting up on an interface of an ad-hoc routing to generating routing table, there are five stages: link detection, neighbor finding, graph decomposition, topology discovery, and routing table computation. Following sections will state the design of these five stages. As a link state routing protocol, node should maintenance some data information to compute routing. For every node of PLSR, there are five information bases to keep the information. Section 4.1 will describes this issue.

4.1 Information Bases

In PSLR there are five Information bases, and they are used to keep some important information including interface association set, link set, neighbor set, SDN set, and topology set. To illustrate their function clearly, some important defines are given as follows:

- PLSR interface: A network device joining in MANET to run PLSR. A MANET node could have several PLSR interfaces, and each assigned an IP address.
- unique address: The unique address is delegate address of the node, and PLSR will use this address to compute routing. For a PLSR node that has only one interface, the address of the only PLSR interface is used as the unique address. For a PLSR node that has multiple interfaces, one of its PLSR interface addresses is chosen as its unique address.
- link: A link is comprised of two interfaces: a local and a remote interface, and these two interfaces are in different nodes. When one of interface of a node has a link to one interface of the other node, the node is said to have a link to another node. When there is a link between two nodes, they are neighbor.

Figure 3 shows the relationship between five information bases and five stages of PSLR. First every routing node collects its own interface information to form “Interface Relationship Data Base”. After the “link detection” stage, a “Link State Data Base” generates in nodes, and this information base keeps link connection pair between nodes. Using “neighbor finding” and “Link State Data Base”, nodes can get their “Neighborhood Data Base”. (Note: In “Neighborhood Data Base”, neighbor is identified by its unique address. But “Interface Relationship Data Base” and “Link State Data Base” use node interface address.) In PLSR there are two kinds of node, SDN and normal node, and they have different functions in PLSR and maintain different topology information to compute routing table. So there should have a SDN Data Base after graph decomposition. To get routing table nodes must know the network topology, and topology discovery will finish this work. When nodes get topology information that they need, they can compute routing.

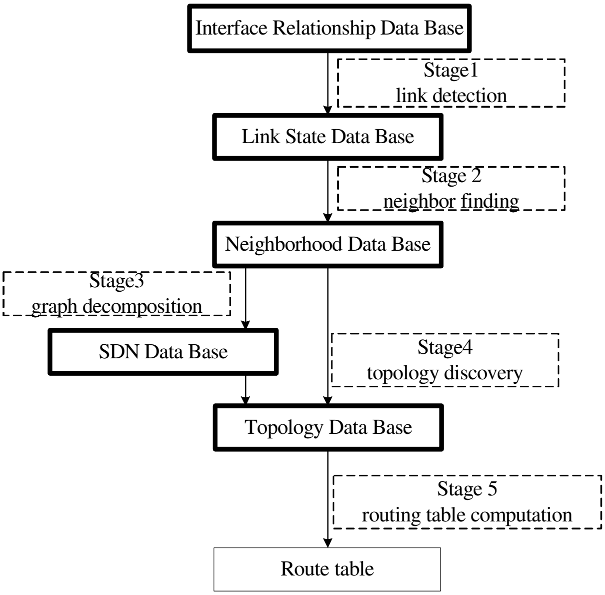


Fig. 3. Flow chart of PLSR

4.2 Link Detection

The "link" in link detection is described by a pair of interfaces: a local and a remote interface. For acquiring the link information between nodes, every node should detect the links between itself and neighbor nodes and maintains several “links” which are called link sets. The link set in Link State Data Base is populated with information on links to neighbor nodes. The process of populating this set is denoted link detection and is performed using Interface association set message exchange, updating a Link State Data Base in each node. The link detection must be processed periodically in order to check the states of link.

4.3 Neighbor Finding

The Link Set keeps the information about the links, while the Neighbor Set keeps the information about the neighbors. There is a clear association between those two sets, since a node is a neighbor of another node if and only if there is at least one link between the two nodes. After the link detection, nodes get link information with other nodes, but routing computation need neighbor information. Neighbor finding populates the Neighborhood Data Base and concerns itself with nodes and node unique addresses, and it transform the relationship of nodes based on interface address to unique address. Likely with link detection, neighbor finding should be periodic, because the link set is updated continuously.

4.4 Graph Decomposition

After all of the nodes know the information of their neighbors, it is the time to decompose the whole topology graph into some sub-graphs for parallel computation. The following specifies a proposed heuristic for graph decomposition.

Step 1. Choosing the nodes that have some big degree (The degree of node *N* is defined as the number of one-hop neighbors of *N*.) as the delegate of a sub-graph. Big degree will connect more nodes with less hops. This will reduce the overload in sub-graphs.

Step 2. Getting rid of the delegate nodes that connect to other delegate nodes using more hops. This is an optional step for optimization.

Step 3. Using the residual delegate nodes as center nodes to extend the sub-graphs according the hops from it.

Figure 2 show an example using above method. There are 4 vertexes whose degree are 4, and the degree is the biggest now. We choose these 4 nodes as SDNs that are solid vertexes. Then use the residual delegate nodes as center nodes to extend 2 hops to form the sub-graphs *A*, *B*, *C*, *D*. The SDNs can only be in one sub-graph. The graph shows that some nodes are in more than one sub-graph, and some nodes are not in any sub-graph. For these nodes, they will be in the same sub-graph with the nodes that establish neighbor relationship firstly.

In section 5 we will give an analysis to describe these two kinds of nodes and give some directions to extend the sub-graphs according the hops in different size of network.

4.5 Topology Discovery

For link state routing protocol, each node must know the whole topology information to allow each node to evaluate routings to destinations in the network. In PLSR node only maintain the topology information of the sub-graph that the node belongs to, and the node will get the routing to other node in same sub-graph using classic link state routing protocol. Meanwhile all SDNs maintain a topology of SDN set in the whole network topology, and they will get the routing between themselves. The topology information is acquired by exchanging neighbor information and SDN information between nodes.

4.6 Routing Table Computation

Each node maintains a routing table that allows it to routing data, destined for the other nodes in the network. The routing table is based on the information contained in the Link State Data Base and the Topology Data Base. The routing table is constantly updated to reflect changes in the topology set and the neighbor set. That is, the routing table will be updated when a neighbor appears and disappears, or when new topology table sets are added, or existing topology set entries remove. All destinations in the network with a known routing will be recorded in the routing table. Shortest path algorithm is adopted during routing evaluation. Whenever the routing table is recalculated, all the routing table entries are initially removed.

5 Simulation and Analyzation

5.1 Graph Decomposition

PLSR is mainly based on graph decomposition, so the method of decomposing the network topology is very important. We have given a three-steps method to decompose graph to some sub-graphs in above section, and there is an analysis to show how to acquire the optimal capability with our method for PLSR.

In paper [13], there is a theorem and its proof that using the decomposition algorithm described as figure 1 model, it is optimal with respect to computational effort to use $O(\sqrt{N})$ processors, and N is the number of nodes. This means that using PLSR to calculate routing in an ad-hoc network environment with N nodes, it is the best to decompose $\sqrt{N}$ sub-graph to get better parallel algorithm efficiency. On the other hand, there is another question using our decomposition method that some nodes are in more than one sub-graph, and some nodes are not in any sub-graph. So how to increase the percent of nodes covered by sub-graph and how to decrease the nodes covered by several sub-graphs simultaneously should be considered.

To resolve these questions and acquire better capability, we make a simulation analysis and give an instruction to decompose graph in PLSR. The simulated network consists of 1000 nodes, and for covering more situations we use four kinds of topology model: Waxman, BA, BA2, GLP. All the topology graphs are generated by universal topology generation tool BRITE [14]. The simulation is according to the steps of our graph decomposition method, except that we consider all the degrees but not a bigger degree. X-axis is the number of degree, Y-axis is the hops, and Z-axis is the percent of nodes covered by all sub-graphs. Figure 4, 6, 8, 10 show the four kinds of topology, and figure 5, 7, 9, 11 show the simulation results.

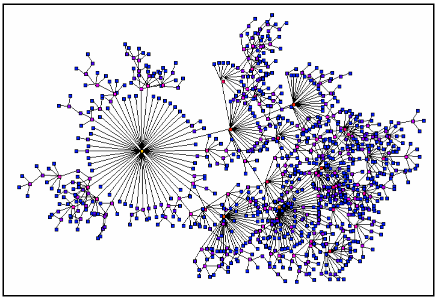


Fig. 4. Topology of BA model

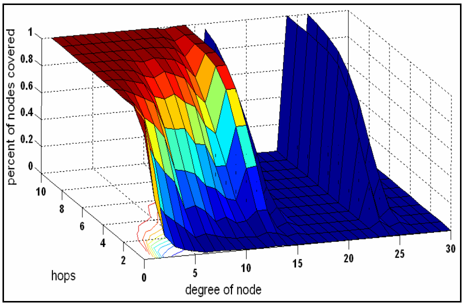


Fig. 5. Simulation result of BA model

From these figures we can conclude following results. Firstly as the reducing of degree and increasing of hops, the percent of nodes covered by sub-graph get to 100%. Secondly there are some breaks in the simulation result figures of BA, GLP, and BA2 model. The reason is that under these topology environments there are no according nodes for some degree number. For Waxman because the degree number is even, so the simulation result graph is very smooth. Finally, though there are more one-hop neighbors for the nodes with bigger degree, the hop number is more important to the percent of covered nodes because bigger degree nodes are rare.

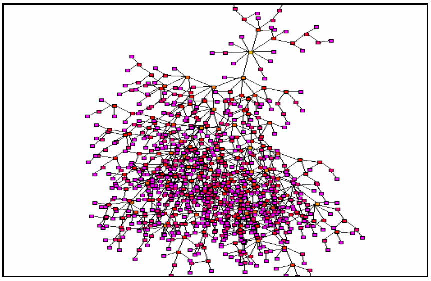


Fig. 6. Topology of Waxman model

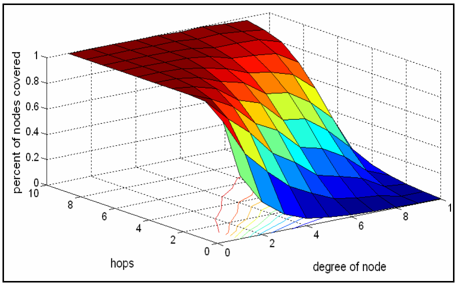


Fig. 7. Simulation result of Waxman model

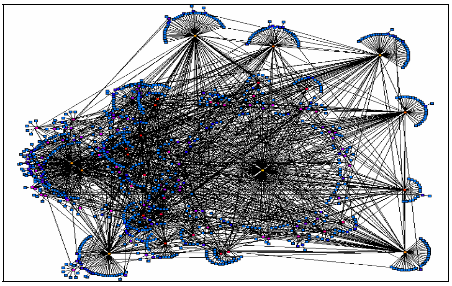


Fig. 8. Topology of GLP model

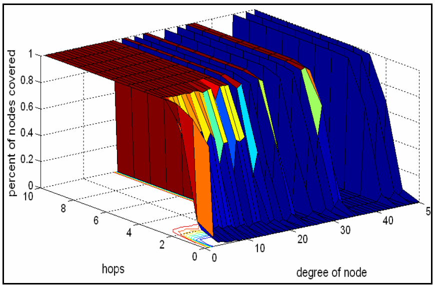


Fig. 9. Simulation result of GLP model

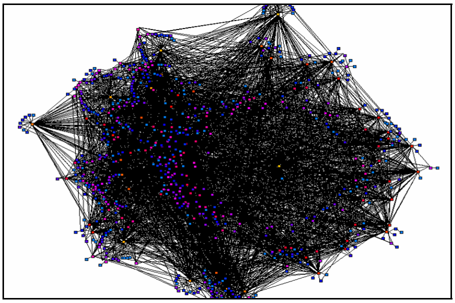


Fig. 10. Topology of BA2 model

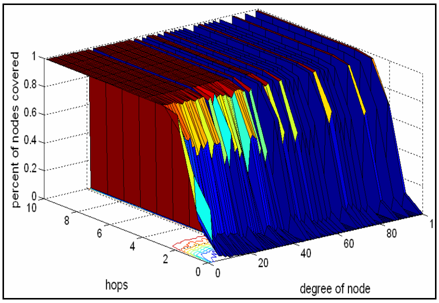


Fig. 11. Simulation result of BA2 model

Table 1 is the optimal choose of degree and hops for PLSR according to simulation results. We have said above that using PLSR to calculate routing in an ad-hoc network environment with N nodes, it is the best to decompose $\sqrt{N}$ sub-graphs to get better parallel algorithm efficiency. So when the number of node with the degree in table 1 is about 31 ($\sqrt{N}$, N=1000), the parallel algorithm will be optimal. Hops are chosen by two principles. It must cover more nodes (We chose the covering percent which is more than 0.9.), and it must be the first hop to get this covering percent. (This can reduce the number of nodes that are covered by multiple sub-graphs.)

In table 1 the first column accords to four kinds of topology models. The second column “Optimal degree” is the degree number for choosing about 31 sub-graphs. The forth column “Optimal hops” is the hops with bigger percent of covered nodes and least hop, and this is according to the two hop choosing principles above. The third column “Not optimal hops” is the pre-hop of the optimal hop, and this hop can’t be chosen optimal hop because of the less percent of covered nodes.

Table 1. Optimal parameter choose

	Optimal degree		Not optimal hops		Optimal hops	
Topology	Degree	Node number	Hops	Covering percent	Hops	Covering percent
Ba	4	38	4	0.78	5	0.911
Wax	5	33	6	0.857	7	0.923
Glp	3	29	3	0.676	4	0.992
Ba2	6	31	3	0.762	4	0.98

The table shows clearly that, in different topology with 1000 nodes, the optimal capability is acquired using degree about five and hops about five with our sub-graph decomposition algorithm.

5.2 Comparison with OSLR and Flooding Model

MANET faces restriction of lacking bandwidth, so the most important issue of MANET routing protocol is how to decrease the overload between nodes. A simple technique to propagate the overload is to let every node retransmit the message when it receives. This technique is called “pure flooding”. In OLSR, the concept of multipoint relaying is used. This concept aims to reduce overload by allowing only a subset of the neighbors, instead of all neighbors of a node to forward control traffic. In PLSR, the technology of graph decomposition is used to decrease the overload. After the graph decomposition, the whole network is divided into many sub-graphs. The overload will be restricted into sub-graph.

In this section we will give a simulation result as figure 12 to compare the overload between pure flooding, OLSR, and PLSR. We use NRL OLSR [15] NS-2 [16] simulation tool that can be configured flooding and OLSR MPR model. Then we have a simple PLSR implementation based on NRL OLSR. Our implementation can only decompose graph average, but this is enough to illuminate how PLSR decrease the overload comparing with OLSR and flooding model. The simulate network consists of 128 nodes, and it can be divided 2, 4, 8, 16, 32, 64 sub-graphs average. We compare the overload inside a sub-graphs using PLSR and the overload between same number of nodes using OLSR and flooding mode. From figure 12 we get following concludes. When there is no sub-graph, PLSR is just as flooding, and OLSR is the best. When there are two sub-graphs, PLSR is better then flooding and worse than OLSR. With the number of sub-graph creasing, there are litter overload for the same number of nodes using PLSR than OLSR. In figure 13, the simulate network consists of 192 nodes, and in figure 14 there are 160.

Besides decreasing the overload, PLSR reduce algorithm time complexity. OLSR uses classic Dijkstra's algorithm to compute routing, and the time complexity is *O(m+nlogn)*. Here *n* is the number of vertices and *m* is the number of edges in the graph. In paper [13], we saw that it is possible to reduce the time complexity by a factor N to *O(NlogN)* by using *N* processors in a parallel fashion. This means that the time complexity will depend on the number of sub-graphs.

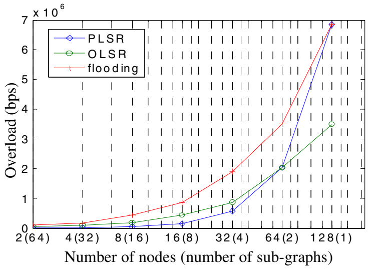


Fig. 12. Overload comparison 1 with OLSR and flooding model

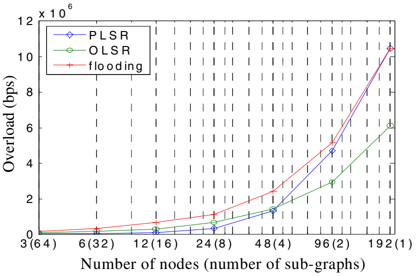


Fig. 13. Overload comparison 2 with OLSR and flooding model

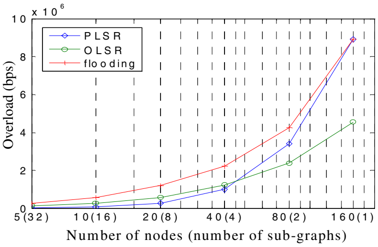


Fig. 14. Overload comparison 3 with OLSR and flooding model

6 Conclusions

This paper presents PLSR that is a new link state routing protocol for MANET. PLSR is a parallel algorithm based on graph decomposition. Using the concept of graph decomposition, the whole network is divided into several sub-graphs, and this will constrain the data exchanging in every sub-graph to reduce the overload between nodes. On the other hand, parallel algorithm can reduce the time complexity. Because of the importance of graph decomposition technology, this paper presents a method to do this work and gives a simulation and analysis to show how to acquire the optimal capability with this method in different topology environment. The simulation and its results will give an instruction to use PLSR.

References

1. Royer, E.M., Toh, C.-K.: A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks. Personal Communications 6(2), 46–55 (1999)
2. Clausen, T., Jacquet, P.: 'Optimized Link State Routing Protocol, RFC 3626 (October 2003)
3. Ogier, R., Templin, F., Lewis, M.: Topology Dissemination Based on Reverse-Path Forwarding, RFC 3684 (February 2004)
4. Berteskas, D.P., Gallagre, R.G.: Distributed Asynchronous Bellman-Ford Algorithm, Data Networks, pp. 325–333. Prentice Hall, Enlgewood Cliffs (1987)
5. Perkins, C., Bhagwat, P.: Highly Dynamic Destination-Sequenced Distance Vector (DTDV) for Mobile Computers. In: Proc. of the SIGCOMM 1994 Conference on Communications Architectures, Protocols and Applications, pp. 234–244 (August 1994)
6. Perkins, C., Royer, E., Das, S.: Ad hoc On-demand Distance Vector (AODV) Routing, RFC 3561, http://www.ietf.org/rfc/rfc3561.txt
7. Park, V., Corson, S.: Temporally-Ordered Routing Alogorithm(TORA) VERSION 1 Internet Draft, draft-ietf-manet-tora-spec- 03.txt, work in progress (June 2001)
8. Johnson, D.B., Maltz, D.A.: Dynamic Source Routing in Ad Hoc Wireless Networks. In: Imielinski, T., Korth, H. (eds.) Mobile Computing. Ch. 5, vol. 353, pp. 153–181. Kluwer Academic Publishers, Dordrecht (1996)
9. Toh, C.-k.: A Novel Distributed Routing Protocol To Support Ad hoc Mobile Computing. In: IEEE IPCCC 1996. Proc. IEEE 15th Annual International Phoenix Conference on Computers and Communications, Phoenix, AZ, USA, pp. 480–486 (March 27, 1996)
10. Haas, Z.J., Pearlman, M.R., Samar, P.: The Zone Routing Protocol (ZRP) for Ad Hoc Networks, Internet Draft, work in progress (July 2002), http://www.ietf.org/proceedings/02nov/I-D/draft-ietf-manet-zone-zrp-04.txt
11. Bertsekas, D.P., Tsitsiklis, J.N.: Parallel and distributed computation. Prentice-Hall, Englewood Cli!s, NJ (1989)
12. Quinn, M.J.: Designing efficient algorithms for parallel computers. McGraw-Hill, New York (1987)
13. Edwin, H., Smith, L.: Parallel Algorithms for Solving Aggregated Shortest-path Problem. Computers & Operations Research, 941–953 (1999)
14. The BRITE webpage, http://www.cs.bu.edu/brite/
15. The NRL OLSR webpage, http://pf.itd.nrl.navy.mil/olsr/
16. The NS-2 webpage, http://www.isi.edu/nsnam/ns/

Analytical Throughput for the Channel MAC Paradigm

Manzur Ashraf, Aruna Jayasuriya, and Sylvie Perreau

Institute for Telecommunications Research,
University of South Australia, SA 5095, Australia
Manzur.Ashraf@postgrads.unisa.edu.au

Abstract. It has been shown analytically [1],[2] that significant performance improvements as compared to existing technologies (e.g., IEEE 802.11) can be achieved in random access wireless networks. In [3] we proposed a fully distributed channel access paradigm based on the opportunistic communication principal called the Channel MAC paradigm suitable for distributed wireless networks such as ad hoc networks. In this paper, we analytically derive the throughput of the Channel MAC. It provides a throughput-limit on the channel-based MAC mechanism in shared multiple access environments without collisions or capturing effects. Both simulation and analytical results reveal possible performance improvement over existing techniques.

1 Introduction

The performance of ad hoc networks with Medium Access Control (MAC) protocols such as IEEE 802.11 falls well short of what is predicted by the theoretical models [1],[2]. This is mainly due to the inability of current MAC protocols to simultaneously take into account the dynamic channel conditions, decentralised channel access and unfairness in access to the common channel [4]. The concept of opportunistic communication has been shown to increase the performances of networks with centralised control [5],[6],[7]. Recently attempts has been made to apply the concept of opportunistic communication to networks with decentralised access mechanisms such as ad hoc and sensor networks [8],[9],[10],[11]. It has to be pointed out that these proposed schemes, although exploiting diversity as a way to determine who has priority for transmission, still use a slotted access system. Hence, in the absence of a central entity which would determine who will transmit based on the "best" channel, collisions will still occur because all nodes with good channel conditions will compete for resources at the beginning of the slot [12],[8],[11].

In [3], authors proposed a new MAC paradigm, called Channel MAC, which exploits the random nature of the fading channel to determine the channel access instances in a decentralised and distributed manner. Simulation results based on a Rayleigh fading channel showed that by using the new MAC paradigm, the network can achieve significantly higher throughput for all channel conditions as

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 375–386, 2007.

compared to 802.11 MAC scheme [3]. In this paper we model the Channel MAC protocol based on two-state channel model and show that the Channel MAC protocol always outperforms 802.11 MAC protocol.

The paper is structured as follows. The objectives and functionality of Channel MAC paradigm are briefly described in section II. This is followed by a description of the system model used in this study. Channel model and key definitions are illustrated here. Section IV describes the proposed analytical throughput model for the Channel MAC study. Simulation model and discussions on the results are given in the next section. Finally section VI concludes this work with our future concerns.

2 Channel MAC Scheme

The objective of Channel MAC paradigm is to use the concept of multiuser diversity to eliminate the scheduling-complexity of nodes in decentralised multihop networks. In Channel MAC, a station pre-arranges the instances at which it will send data-packets, based on the predicted channel gain between the node and the intended receiver and a channel gain threshold P_{th} for transmission. When the predicted channel gain goes above the P_{th} threshold the corresponding station potentially starts transmission. However, before sending data, a node will sense whether the channel is busy or not. If the medium is idle, i.e no other node is currently transmitting, the node starts transmission and continues it until the channel gain goes below the P_{th} threshold (i.e the channel goes into a fade). Number of packets transmitted during a good channel period depends on the packet size and the duration of the good channel period. If any other channel becomes good during transmission, the corresponding node will sense the channel being busy and will not transmit. It should be noted here that the carrier-sensing threshold of the nodes is set to a much lower value than the receiving threshold. Hence the transmitters should sense the medium busy even if the channel gain between a transmitter and an interfering node is low.

Given that each transmitter-receiver pair has an independent fading channel, the probability of two or more channels crossing the transmission threshold on a positive slope exactly at the same time is assumed to be negligible. However, due to finite propagation and sensing delay of the nodes, collision can occur, decreasing the throughput. We observe that the effects of collisions on the throughput is no worse than what is observed in other sensing based MAC protocols. It should be noted that the Channel MAC does not rely on random backoff mechanism to randomise the access to the shared medium. Instead, Channel MAC uses the random nature of channels between different node pairs to randomise the channel access. The decision to transmit is taken at each node without explicit knowledge of channel gain between other nodes in the neighbourhood, hence the fully distributed nature of the scheme.

3 System Model

Let us define a neighbourhood of $2n$ nodes, where $N_T \in (1, 2, \dots, n)$ are the transmitters and $N_R \in (1, 2, \dots, n)$ are the receivers. For symmetry let us assume that each transmitter $i \in N_T$ is communicating with receiver $i \in N_R$.

3.1 Channel Model

We consider a simple two-state channel model. It has either a non-fade state, "ON" with gain 1 or a fade state, "OFF", with gain 0. Suppose the distribution of each non-fade duration, $l_i (i \in \Re)$ is $f_n(x)$ with mean l. $\bar{l}$=min(l_i); for all i. Now, in light of the discussion of [13], the inter-arrival point distribution of one channel (instance at which the channel becomes good) can be approximated in general as a variant of the sum of Weibull distribution,

$$f(x) = \sum_z \lambda \beta_z (\lambda(x - \bar{l}))^{\beta_z - 1} exp((-\lambda(x - \bar{l}))^{\beta_z}) \tag{1}$$

where β_z is the shape parameter. The distribution depends on $l_i \in L$. Furthermore, $x > \bar{l}$ and $\bar{l} > 0$, i.e. any two arrival points of a channel are separated by a positive value.

For simplicity, we assume that the non-fade duration, termed Average Non-Fade Duration (ANFD), l, is constant, after which the channel goes into a fade with exponentially distributed fade duration as shown in figure 1. The instantaneous (i-th) idle time of $\bar{n}$ channel, denoted as $\Theta_{\bar{n}i}$, is an exponentially distributed random variable with the mean Θ; where $\bar{n} \in n, i \in \Re$. Hence probability of good channel, p, can be calculated as follows:

$$p = \frac{l}{l + \Theta} \tag{2}$$

We assume that all the channels in the network have the same p value.

3.2 Propositions

When the number of users in the network is 1 (this system is termed 1-user Channel MAC), the resulting transmission pattern of the network is identical to the channel model.

Proposition 1: In 1-user Channel MAC, the arrival point (start of transmission) process is approximated by a Renewal process.

Proof: The inter-arrival distribution can be considered as a shifted exponential of the form $f(x) = \lambda e^{-\lambda(x-l)}; (x \geq l)$ where $1/\lambda$ is the mean of the inter-arrival time. It follows from [14] that the arrival point process is approximated as a renewal process. □

Using $\beta = 1$ and $z = 1$ (a single Weibull distribution) in Equation 1, we get the shifted exponential distribution of the inter-arrival process of 1-user Channel

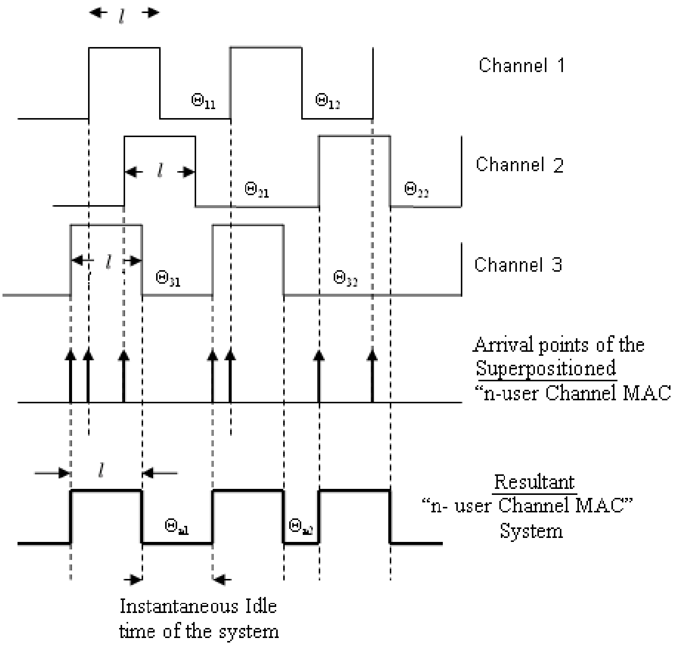

l
Channel 1
Θ11
Θ12
Channel 2
l
Θ21
Θ22
Channel 3
l
Θ31
Θ32
Arrival points of the Superpositioned "n-user Channel MAC
l
Resultant "n- user Channel MAC" System
Θs1
Θs2
Instantaneous Idle time of the system

Fig. 1. Two-state channel model

MAC. We define "Expected period of 1-user Channel MAC", T_p as the expected renewal period [14] of the process. T_p can be expressed in terms of the number of arrival points per unit time period (i.e. Level Crossing Rate, r) as follows:

$$T_p = 1/r = 1/\lambda + l = t + l \tag{3}$$

where $t = 1/r - l$ is the expected idle time for 1-user Channel MAC as shown in Figure 2.

Proposition 2: The shifted exponential distribution function (defined in Proposition 1) results in a non-Poisson renewal arrival process.

Proof: The hazard rate of the distribution, $r(x)$, is λ when $x \geq l$. But if $x < l$, $r(x) = 0$. It follows that the arrival process is a non-Poisson renewal process. □

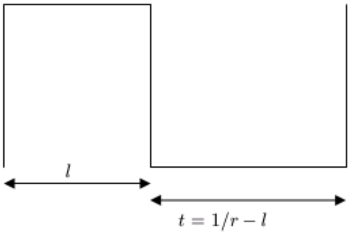


Fig. 2. Expected period of 1-user Channel MAC

Arrival points of n-user Channel MAC. We define the "Superpositioned n-user Channel MAC" as the superposition ([14], pp 101-104) of arrival points of n independent channels. We assume that, at each instance, exactly one channel becomes good (i.e transition from OFF to ON). Corresponding station then can transmit data given that no one else is transmitting at that instance. Following the operation of the Channel MAC scheme, we can identify the transmission periods and idle periods of the network with n users, which we term as "Resultant n-user Channel MAC" system.

Note the difference between *Resultant* and *Superpositioned* n-user Channel MAC. In Resultant n-user Channel MAC, the number of arrival points (i.e transition from OFF to ON) cannot be greater than the number of arrival points in the Superpositioned n-user Channel MAC. It is due to the fact that some of the arrival points of the Superpositioned n-user system may not contribute to throughput in Channel MAC operation as they may occur while another node is transmitting.

We further assume that in Superpositioned n-user Channel MAC, arrival points of individual channels are "sparse". i.e., in any particular $\bar{A}$ set of arrival points occurring in a random time-interval, there will be with high probability, at most one point from each process. In addition, no arrival points from one channel dominates over others. Hence equal number of arrival points from different channels should be present in any random interval.

4 Modelling Channel MAC Throughput

4.1 Superposition of Point Processes

It is known that the superposition of two independent Renewal processes is itself a Renewal process iif all three processes are Poisson [15]. Since the arrival points of 1-user Channel MAC does not constitute a Poisson process (by Proposition 2), the Superpositioned n-user Channel MAC is not a Renewal process either. To simplify the analysis, practical applications such as the superposition of arrival processes in a "Single server queuing model" consider approximation based approaches where the superimposed point process is approximated as a renewal process [16].

It is also well-known that the superposition of independent and uniformly sparse processes converges to a Poisson process as the number of processes and the sparseness increase. Such convergence results were first examined by Palm, in 1943 and Khinchin in 1955 under rigid assumptions [17]. A general Poisson limit theorem for independent superpositions was obtained by Grigelionis in 1963 [18]. This theorem states that if points of each individual processes are (a) suitably sparse and (b) no one process dominates the rest, then the distribution of the point process is close to Poisson. Corresponding results for dependent (mixing) Point processes with Poisson and compound Poisson process in the limit can be found in [19]. All these works conclude that a Poisson process is often a good approximation for a superposition process if many processes are being superposed.

Based on the discussions we state the following proposition:

Proposition 3: The arrival points of the Superpositioned n-user Channel MAC converges asymptotically to a Poisson Point process as per our assumptions.

4.2 Expected Idle Time of Resultant *n*-User Channel MAC

Proposition 4: In Poisson Point process, if n number of arrival points occur in an interval T, the expected delay of the first arrival point in T is $\frac{1}{n+1} \times T$.
Proof: According to ([14], pp. 46; [20], pp. 125) the arrival points of a homogeneous Poisson process with constant rate of arrival are independently and uniformly distributed over the interval. In other words, if n i.i.d. uniform random variables on $[0, \hat{t}] \equiv T$ are arranged in increasing order, they represent n successive occurrence times of a Poisson process. The average spacing of them is $\frac{\hat{t}}{n}$. □

It can be observed that the the expected idle time, $E[I]$ of the system decreases with the increasing number of channels. As per our assumptions the Superpositioned n-user Channel MAC is approximated as Poisson Point process (see Proposition 3). Then we derived the expected delay of first arrival point for a Poisson arrival process (see Proposition 4). Based on Proposition 3 and 4, we can derive $E[I]$ for the Resultant n-user Channel MAC as follows:

After a successful transmission by any channel, independent 0,1,...,∞ arrivals(s) may occur during immediate next fade duration, t, of that channel. Hence the expected idle time is the the weighted average of idle times for all possible number of arrival points.

$$E[I] = \sum_{i=0}^{\infty} P_i I_i \tag{4}$$

where P_i is the probability that i arrival occurs in t and I_i is expected delay of first occurrence of arrival point in this case. P_i is Poisson distributed as per our assumptions.

$$\begin{aligned} E[I] &= \sum_{i=0}^{\infty} \frac{(nrt)^i}{i!} e^{-nrt} \frac{1}{i+1} t \\ &= \frac{1}{nr} \left(1 - e^{-nrt}\right) \end{aligned}$$

where $t = 1/r - l$.

4.3 Analytical Throughput Measurement

The expected period of arrival point process for the Resultant n-user Channel MAC, $\hat{T_p}$ is the summation of the expected duration of successful transmission, l and expected idle time, $E[I]$. The average channel utilization or throughput, S

of Channel MAC is given by the ratio of l to the expected period of the Resultant n-user Channel MAC [21].

$$\begin{aligned} S = \frac{l}{\hat{T}_p} &= \frac{l}{l + E[I]} \\ &= \frac{l}{l + \frac{1}{nr}\left(1 - e^{-nrt}\right)} \end{aligned} \tag{5}$$

where $t = 1/r - l$.

5 Simulation

5.1 Simulation 1: Fixed *l* and Exponential Fade Duration

In this section we simulate the performance of Channel MAC based on the channel model (fixed l and exponentially distributed fade duration with mean $1/r - l$) discussed in section 3.1. The simulation approach we used is to generate n independent channels with same l and average fade duration $1/r - l$. When one or more channel “ON” periods overlap only the first channel to go to “ON” state after a non-zero idle period contributes to the throughput.

We have not considered the collision in analytical form. But in the simulation, we perceive the collisions as follows: if more than one start points of the overlapped non-fade durations of channels are same, we ignore all of them as it indicates a collision. The next immediate non-fade duration of the channel gets the opportunity to transmit for simplicity in our simulation. Practically, parameters such as packet-length, control packets, data-rate, etc, determine the next arrival point which will contribute to the throughput after a collision. The full impact of packet collisions will be observed by the discrete event simulation of the Channel MAC. By a Monte Carlo simulation we derive the throughput of the system over a number of simulation runs. Furthermore, we assume same p for all the stations in multiple node simulations.

5.2 Simulation 2: Rayleigh Fading Model

In the second simulation approach, we generate a set of “ON” and “OFF” intervals based on a Rayleigh distribution. The channel gain characteristics (i.e the times where channel gain is greater than the threshold) are Rayleigh distributed. p, which is equivalent to the probability that the channel gain H_i, is above a certain threshold, $H_{\hat{T}}$ is given by

$$p = \exp^{\left(-\frac{H_{\hat{T}}^2}{h_0^2}\right)} \tag{6}$$

where h_0 is the average value of fading.

In the simulation, for a given p value we derive the channel gain threshold, $H_{\hat{T}}$. Then we generate a channel model, covering a time period $\hat{T}$, in the form of

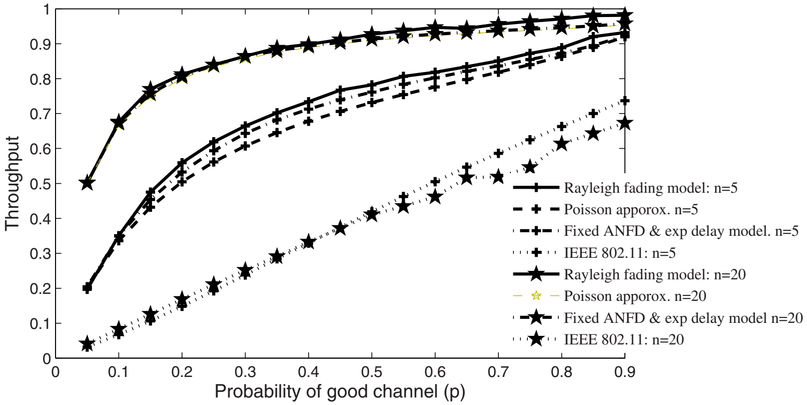


Fig. 3. Throughput vs. p for different number of stations

a set of time intervals, $\mathbf{\Lambda} = \{\lambda_1, \lambda_2, \ldots, \lambda_i, \ldots\}$, where the channel gain is above the threshold $H_{\hat{T}}$. These $\mathbf{\Lambda}$ time periods are the transmission intervals of a node when the probability of good channel is p. For n nodes, n sets of independent $\mathbf{\Lambda}$ time intervals, were generated. In case of overlapping transmission intervals from different nodes, only the first transmission interval in the overlapping group contributes to the throughput. We assume same p for all the stations in multiple node simulations.

Network throughput of Channel MAC for different probabilities of good channel is presented in Figure 3. Performance of IEEE 802.11 under fading channel conditions based on [22] are also shown in this figure.

5.3 Results Comparison

In spite of the slight variations in results (particularly for lower number of nodes), it can be noted that Channel MAC outperforms 802.11 for all p values in all cases. It can be noted that for higher number of nodes Channel MAC achieves higher throughput at lower p values, increasing the potential operating range. Furthermore the total throughput of the network will continue to increase with increasing number of nodes due to multiuser diversity, contrary to the performance of most other medium access control protocols (i.e., IEEE 802.11). For example for $n = 5$, throughput of the system is greater than 0.8 for $p > 0.6$, while for $n = 20$, same level of throughput can be achieved for $p > 0.2$. Comparing to a typical operating setting of IEEE 802.11 MAC where p=0.9 and $n = 5$, Channel MAC outperforms the corresponding throughput by 26% with the same number of nodes. It grows to 43% when $n = 20$ for both systems.

In Figure 4, The throughput vs number of stations in the Channel MAC and the IEEE 802.11 is shown at $p = 0.1$ and 0.85. It can be noted that the discrepancy between simulation and analytical model results decreases at lower p.

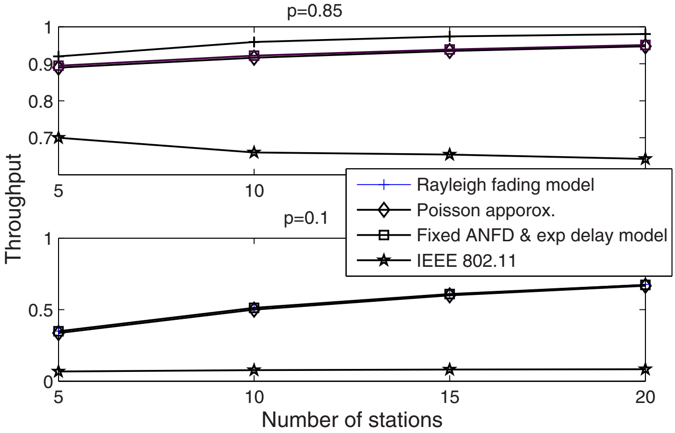


Fig. 4. Throughput vs. number of stations for $p = 0.1$ and $p = 0.85$

Furthermore, the simulation results should approach to the Poisson approximation with the increasing number of nodes. A detailed analysis of this discrepancy is given in Appendix A.

6 Conclusion

In this paper we evaluate analytically the Channel MAC paradigm. The analytical and simulation results presented show that Channel MAC can achieve higher throughput than IEEE 802.11 in distributed wireless networks. Moreover, the throughput in channel MAC scheme increases with increasing number of nodes, due to the multi user diversity of the system. Drawbacks of this scheme are the bandwidth required to exchange channel information between transmitter-receiver pairs and the added processing power required to predict channel conditions.

The analytical model accurately captures the behaviour of Channel MAC protocols for the two-state channel model presented in the paper. Furthermore through the comparison of analytical results to a channel model based on Rayleigh fading we have shown for large values of n the analytical model presented here, closely matches the performance of Channel MAC protocol in Rayleigh fading channels. In future work we aim to develop an analytical model to capture more general channel models.

Acknowledgements

The authors are grateful to Professor Lars Rasmussen, University of South Australia for useful feedback on the paper and general discussions.

References

1. Gupta, P., Kumar, R.: The capacity of wireless networks. IEEE Transactions on Information Theory 46(2), 388–404 (2000)
2. Grossglauser, M., Tse, D.: Mobility increases the capacity of ad-hoc wireless networks. IEEE/ACM Transactions on Networking 44(4), 477–486 (2002)
3. Ashraf, M., Jayasuriya, A., Perreau, S., Rasmussen, L.: Channel mac: A novel medium access control paradigm for wireless ad hoc networks. In: Australian Telecommunication Networks and Applications Conference, pp. 404–408 (December 2006)
4. Sagduyu, Y., Ephremides, A.: The problem of medium access control in wireless sensor networks. IEEE Wireless communications 11(6), 44–53 (2004)
5. Goldsmith, A., Varaiya, P.: Capacity of fading channels with channel side information. IEEE Transactions on Information Theory 43(6), 1986–1992 (1997)
6. Tse, D., Hanly, S.: Multi-access fading channels: Part i: Polymatroid structure, optimal resource allocation and throughput capacities. IEEE Transactions on Information Theory 44(7), 2796–2815 (1998)
7. Hanly, S., Tse, D.: Multi-access fading channels: Part ii: Delay-limited capacities. IEEE Transactions on Information Theory 44(7), 2816–2831 (1998)
8. Qin, X., Berry, R.: Exploiting multiuser diversity for medium access control in wireless networks. In: Proceedings of INFOCOM, vol. 2, pp. 1084–1094 (2003)
9. Srihari, A., Lang, T.: Exploiting decentralized channel state information for random access. IEEE Transactions on Information Theory 51(2), 537–561 (2005)
10. Venitasubramaniam, P., Adireddy, S., Tong, L.: Opportunistic aloha and corss layer design for sensor networks. In: Proceedings of MILCOM, pp. 705–711 (October 2003)
11. Zhao, Q., Tong, L.: Distributed opportunistic transmission for wireless sensor networks. In: Acoustics, Speech, and Signal Processing (ICASSP 2004), vol. 3, pp. 833–836 (May 2004)
12. Knopp, R., Humblet, P.: Information capacity and power control in single-cell multi-user communications. In: Proceedings of IEEE International Conference on Communications, pp. 331–335 (June 1995)
13. Feldmann, A., Whitt, W.: Fitting mixtures of exponentials to long-tail distributions to analyze network performance models. In: Proceedings of INFOCOM, pp. 1096–1104 (1997)
14. Cox, D., Isham, V.: Point Processes. Chapman and Hall, Australia (1980)
15. Cinlar, E.: Superposition of point processes. Stochastic point processes: statistical analysis, theory and applications, pp. 549–606 (1972)
16. Whitt, W.: Approximating a point process by a renewal process, i: Two basic methods. Operations Research 30(1), 125–147 (1982)
17. Schuhmacher, D.: Estimation of distances between point process distributions, PhD thesis, University of Zurich (2005)
18. Grigelionis, B.: On the convergence of sums of random step processes to a poisson process. Theory of Probability Applications 2(8), 177–182 (1963)
19. Banys, R.: On superpositions of random measures and point processes, NY. Mathematical statistics and probability theory, Lecture notes in Statistics, vol. 2, pp. 26–37 (1978)
20. Mieghem, P.: Performance analysis of communications networks and systems. Cambridge University Press, Cambridge (2006)

21. Kleinrock, L., Tobagi, F.: Packet switching in radio channels: Part 1- carrier sense multiple access modes and their throughput-delay characteristics. IEEE Transactions of Communications 23(12), 1400–1416 (1975)
22. Pham, P., Perreau, S., Jayasuriya, A.: New cross layer design approach to ad hoc networks under rayleigh fading. IEEE Journal on selected areas in communications 23(1), 28–39 (2005)

Appendix A: Difference in Poisson Approximation and Simulation Results of Channel MAC Throughout

The expected throughput between the Poisson approximation and that of the simulation 2 differs as observed in figure 3 and 4. It is related to the discrepancies between the independence measure of Poisson approximation and that of the shifted exponential assumption of the channel model.

A shifted exponential distribution was assumed for the inter-arrival process of 1-user Channel MAC. Its density is

$$f(x) = \begin{cases} \hat{L}e^{-\hat{L}(x-d)} & x \geq d \\ 0 & otherwise \end{cases} \tag{7}$$

where $\hat{L}^{-1}$ is the mean of the exponential variable and d is constant. The two parameters $\hat{L}$ and d are related to the mean μ and variance σ^2 of the shifted exponential density by

$$\mu = \hat{L}^{-1} + d; \sigma^2 = \hat{L}^{-2} \tag{8}$$

Since $d = l$ and $\mu = 1/r$ in 1-user Channel MAC, it follows that the Co-efficient of Variation (CV) of the inter-arrival process, c_a is

$$c_a = \frac{\sigma}{\mu} = \frac{1/\hat{L}}{1/\hat{L} + l} = 1 - p \tag{9}$$

In case of the superpositioned n-user Channel MAC, the mean of the exponential variable L and CV, c_{sup} are related to the component counterparts by following equations [16]:

$$L = \sum_{i=1}^{n} L_i = nL_i; c_{sup} = \sum_{i=1}^{n} (\frac{L_i}{L}) c_i \tag{10}$$

where L_i and c_i are the mean of the exponential variable and CV of the ith component process, $i \in n$. As we consider same L_i values for all the channels, it follows that

$$c_{sup} = c_i = 1 - p \tag{11}$$

In simulation 2, due to approximation error in the Rayleigh fading model of the n-user Channel MAC, we get the non-linearity pattern in figure 5. It differs from the linear function of equation 9. Therefore, the simulation result approaches to the Poisson approximation at lower p values.

Again the mean arrival rate of the superpositioned n-user Channel MAC, $\mu_s = 1/nr = 1/L + d_s$.

$$\therefore d_s = \frac{1}{n}\left(\frac{1}{r} - \frac{1}{L_i}\right) = \frac{l}{n} \tag{12}$$

$d_s = 0$ results in a Poisson arrival point process. Hence the absolute value of d_s indicates a measure of independence. In simulation 2, the distribution of the arrival point process approaches to Poisson as $d_s \to 0$. For a fixed value of n, l increases with p. d_s is also increased evidently. Hence, for lower values of n (=5,10, etc e.g.,), the simulated results deviate more from the Poisson approximation as p increases. Consecutively, as n increases, $d_s \to 0$, indicating the distribution of the arrival process approaches to Poisson.

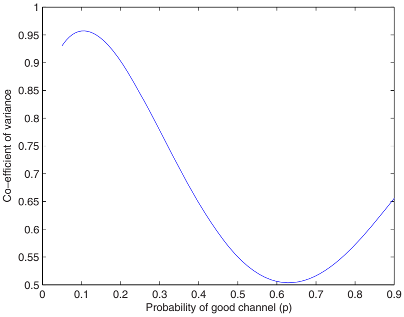


Fig. 5. approximate c-p curve pattern (smoothed) for 1-user Channel MAC by Simulation 1

An Adaptive Transmission Control Scheme Based on TCP Vegas in MANETs

Liu Hongfei[1,2], Sheng Hongyan[3], Li Lijun[4], Yang Zuyuan[1], and Huang Xiyue[1]

[1] Automation Academy, Chongqing University,
400044 Chongqing, China
[2] Chongqing Communication College,
400035 Chongqing, China
[3] Institute of Physics & Electronic Engineering, Ludong University,
264025 Yantai, China
[4] School of Mathematics and Sciences, Chongqing Institute of Technology
400050 Chongqing, China
{cqtxxy123@126.com,mchlong@126.com,llj.liu@cqit.edu.cn,yzy7704@163.com,xyhuang@cqu.edu.cn}

Abstract. Traditional transmission control protocol reduces its performance by misinterpreting mobility losses due to node motion as congestion losses in wireless mobility network. While it has been shown that TCP Vegas provides better performance compared to TCP Reno, studies have identified various issues associated with the protocol in mobile ad hoc networks. This paper proposed adaptive congestion control scheme based on Traditional TCP. The key idea of our approach is to monitor permanently the TCP flow and record useful data to infer the current state of mobility ad hoc networks when packet losses are perceived, distinguish losses packets between bit error and congestion based on fuzzy logic theory, then based on wireless network link state, such as bit error, network congestion, or rerouting, Appropriate congestion control algorithm be choose to overcome limitations. Our experiments show that the scheme is able to avoid oscillation of TCP sender congestion window, and obtain more throughput than traditional transmission control protocol.

Keywords: mobile ad hoc network, adaptive transmission control protocol, congestion control, error detection mechanism.

1 Introduction

Mobile Ad Hoc network can form network temporarily without using existing infrastructure. In particular, it is fast to configure the network and is inexpensive for construction since it has relatively less restriction due to movement of network terminal and does not require contact point with the center [1]. In addition, Ad Hoc network can be used in several areas such as vehicular ad hoc network in intelligent transportation systems, disaster in military communication network and so on. Since mobile Ad Hoc network communicates using wireless media, it has restriction of communication such as interference of signal, noise and data loss against environment.

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 387–396, 2007.

Transmission Control Protocol, which is a reliable transport protocol designed for the Internet, has been well studied and tested over the years. Congestion control is a distributed feedback algorithm to allocate network resources among competing users. To estimate the available bandwidth in the network, TCP Reno uses packet loss as an indicator for congestion. Its congestion window will be increased until packet loss is detected, at which point the congestion window is halved and then a linear increase algorithm will take over until further packet loss is experienced.

However, the performance of TCP is severity degraded due to well-known problems characteristic of wireless transmissions [2] [3] [4]. This is because TCP's flow and congestion control mechanisms are based on the assumption that packet loss is an indication of congestion. While this assumption holds in wired networks, it does not hold in wireless mobile ad hoc scenarios, such losses may occur not only by congestion but also due to both the typically high Bit Error Rate (BER) of the wireless channels and link interruptions by mobility. Channel errors induce TCP to mistakenly reduce its transmission rate by halving its congestion window (cwnd) [4]. Mobility induced losses (due to link interruption) may lead TCP to incredibly long periods of inactivity due to its exponential backoff mechanism. In both cases the TCP throughput will be impaired.

TCP Vegas is a new design for TCP that was introduced by Brakmo et al. [5]. With a fundamentally different congestion avoidance scheme from that of TCP Reno and claimed that TCP Vegas achieves 37 to 71 percent higher throughput than TCP Reno. Ahn et al. [6] have evaluated the performance of TCP Vegas on their wide area emulator and shown that TCP Vegas does achieve higher efficiency than TCP Reno and causes much less packet retransmissions.

At present, fuzzy logic is becoming popular especially with the use of IP networks [4] [8] [9]. They have the advantages to perform non-linear input and output mapping from training data and to express the ambiguity of knowledge in linguistic terms.

This paper presents an adaptive TCP Vegas algorithm in MANET that is called TCP Vegas_M (TCP Vegas MANET). The key idea of our approach is to monitor permanently the TCP flow and record useful data to infer the current state of ad hoc networks when packet losses are perceived, distinguish losses packets between bit error and congestion based on fuzzy logic theory, then based on ad hoc network state, such as bit error, network congestion, or rerouting, we propose modification to the congestion avoidance mechanism to overcome issues of the TCP Vegas in MANET.

The rest of this paper is organized as follows. Section 2 discusses related work. Section 3 briefly describes the algorithm and performance of TCP Vegas. Section 4 describes previous algorithms for distinguishing between bit error and congestion induced losses, and introduces TCP Vegas Mobility algorithm, a novel algorithm for improving system throughput in MANET. Section 5, we report the results of simulation carried out over various environment. Finally, section 6 summarizes and concludes.

2 Related Work

Over the last few years, there are many modifications and new solutions have been proposed to improve TCP's performance. Existing approaches may be classified into two classes: Network oriented [11,12] and end-to-end [13,14] solutions.

Literature [12] proposed ATCP is based on Network oriented, which deals with the problems of high BER, route failures, network partitioning and multipath routing. A thin layer called ATCP is inserted between TCP and IP layers. The ATCP layer monitors TCP state and the state of the network and takes appropriate action. The ATCP's four possible states are: Normal, Congested, Loss, disconnected. When ATCP sees that three duplicate ACKs have been received, it considers it a channel loss and only transmits the unacknowledged segments. Congestion is detected by ECN message. In case of temporary network partitioning, the ATCP receives an ICMP "Destination Unreachable" message. Hence, it puts the TCP sender in the persist state, sets TCP's congestion window into one and enters itself in the disconnected state. TCP periodically generates probe packets until it start receives their ACKs. This removes TCP from persist mode and moves ATCP back into normal state. An obvious limitation of this approach is that these techniques need to be deployed at every node [10].

ADTCP [13] is an end-to-end approach, which is based on the use of multi-metric joint identification in order to detect different network states in order to take the appropriate reaction. They introduced four different metrics (IDD, STT, POR, PLR) to be measured. The first metric IDD reflects the congestion level along the forwarding delivery path. The second metric is STT, which is also used to detect network congestion. The other two metrics are used for non-congestion state identification. POR is intended to indicate a route change event while PLR is used to measure the intensity of channel error. Upon each packet arrival at the receiver, it calculates the above four metrics, estimate the network state and send the state information to the sender with every ACK packet so the sender can take the suitable action.

In literature [4], the authors also make use of Fuzzy Logic theory, for distinguishing between bit error and congestion induced losses, using RTT values as input variables. By using fuzzy logic, the continuous and imprecise behavior of the information can be handled without the necessity of arbitrary rigid boundaries. Besides, it is low processing demanding. This renders fuzzy logic quite suitable for evaluating RTT values where imprecision and uncertainties are effectively present and the processing requirements must be as low as possible. In this paper, which is an extension of [4], we focus on our approach for enhancing the TCP congestion avoidance mechanism by using fuzzy logic to distinguish network states. TCP recovery strategy is left for future work.

3 TCP Vegas and Issues

This section briefly reviews the innovations of TCP Vegas with respect to TCP Reno that are most relevant to developing the throughput model. One of the most important aspects is the Vegas congestion avoidance mechanism, which differs significantly from TCP Reno [2]. While TCP Reno uses consider the loss of packets as a signal that there is congestion in the network. In fact, Reno needs to create losses to find the available bandwidth of the connection. In contrast, the goal of Vegas is to pro-actively detect congestion in its incipient stages, and then reduce throughput in an attempt to prevent the occurrence of packet losses. The goal of the Vegas congestion avoidance algorithm is to keep this number within a fixed range defined by two thresholds. TCP Vegas adjusts the congestion window size as follows [15]:

$$cwnd(n+1) = \begin{cases} cwnd(n)+1 & diff < \alpha \\ cwnd(n) & \alpha \le diff \le \beta \\ cwnd(n)-1 & diff > \beta \end{cases} \tag{1}$$

Where: diff=[cwnd(n)/baseRTT]-[cwnd(n)/RTT].
expected_rate=cwnd(n)/ baseRTT,
Where cwnd(n) is the current congestion window size and baseRTT is the minimum RTT of that connection.
actual_rate=cwnd(n)/RTT.
Where RTT is the actual round trip time.

α and β are throughput threshold parameters whose values are usually set as 1 and 3, respectively.

Another feature of Vegas is its modified slow-start behavior and recovery algorithms. This paper we mainly discuss the congestion avoidance mechanism in our modification algorithm and slow-start and congestion recovery mechanism of TCP Vegas-M are the same as that of Vegas. For more detailed on TCP Vegas, refer to [5].

Although many researched results shown that TCP Vegas does achieve higher efficiency than TCP Reno, causes fewer packet retransmissions, several issues have also been identified with the protocol [7]. Firstly, TCP Vegas can't distinguish packet losses between network congestion and bit error, so it isn't suited to ad hoc network. Secondly, TCP Vegas doesn't handle rerouting well. In Vegas, the parameter baseRTT denotes the smallest round-trip delay and is used to measure the expected throughput. When the route is changed, the RTT of a connection can change. Vegas is not usually affected if the new route has a smaller RTT, since baseRTT will be updated dynamically. But when the new route has a longer RTT, the connection will not be able to deduce whether the longer RTTs experienced, TCP Vegas assumes that the increase in RTT is because of congestion along the network path and hence decreases the congestion window size. This issue will degrade TCP throughput and affect network performance.

La et al. [7] proposed a modification to Vegas to counteract the re-routing problem. Their modification assumes that any lasting increase in RTT is a sign of re-routing, for the first K packets, the mechanism is the same as TCP Vegas. However, subsequently, the source keeps track of the minimum RTT of every N packet. If this minimum RTT is much lager than the baseRTT for L consecutive times, the source updates its baseRTT to the minimum RTT of last N packet and resets the congestion window based on this new baseRTT. Their reasoning for this modification is that since the increase in RTT forces the source to decrease the congestion window, the increase in RTT comes mostly from the propagation delay of the new route and not from the congestion. However, this mechanism adds two more parameters to control the update process of baseRTT, and finding the appropriate value for K, N, L and others remains an open issue.

4 Design of TCP Vegas Congestion Control Mechanism Based on Fuzzy Logic Theory

4.1 Base Idea of FEDM

Ruy de Oliveira et al [4] not only make use of fuzzy logic theory for distinguishing between bit error and congestion induced losses, but also define the *NH* (Number of Hops) and *RR* (RTT increase Rate) blocks for detecting the number of hops in the end-to-end connection and steep increases in the RTT measurements, solving the rerouting problem in ad hoc networks.

To distinguish bit error and network congestion, literature [4] propose Fuzzy-based Error Detection Mechanism (FEDM) Fig. 1 depicts the general architecture of FEDM.

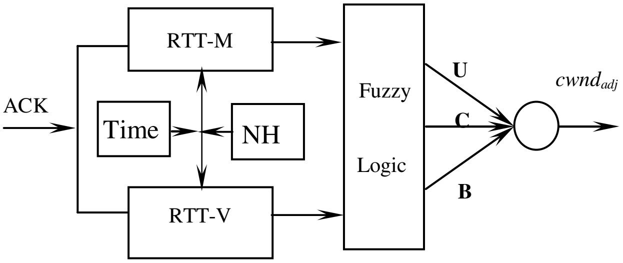


Fig. 1. General Architecture of FEDM

The input variables of the fuzzy engine are defined as the RTT mean *t* in equation (1) and the RTT variance *δt* in equation (2, 3). The output of each fuzzy rule in FEDM is assigned to a corresponding output fuzzy set. The corresponding fuzzy linguistic variables are C (Congestion), U (Uncertain) and B (Bit Error). As congestion has priority over bit error, the C variable covers also conditions of simultaneous congestion and bit error constraints.

$$t = \frac{1}{n}\sum_{i=1}^{n} t_i \tag{2}$$

$$\delta = \frac{1}{n}\sum_{i=1}^{n} (t_i - t) \tag{3}$$

For computing simplicity and better control of the spread of the curves, [4] use Gaussian membership functions for the input memberships. The universe of the fuzzy input variables *t* and *δt* are divided into three fuzzy sets as shown in Fig. 2(a,b). The fuzzy linguistic variables used are S (Small), M (Medium) and L (Large). The specific fuzzy rules refer to literature [4].

To solve the rerouting problem in MANET, Ruy de Oliveira et al propose to use the Time To Live (TTL) field within the IP header for identifying the mentioned number of hops. This demands a simple interoperation between transport and network layer protocols and the use of either an IP or TCP option inside the packet header.

Upon packet receipts, the NH block may use the current TTL along with the TTL sent to compute the exact number of hops crossed by the connection.

Ruy de Oliveira et al have proposed a simply method to distinguish network congestion and bit error, and solve rerouting problem, but didn't present appropriate TCP algorithms over MANET. So, in next subsection, we present modification TCP Vegas algorithm based on FEDM and Vegas.

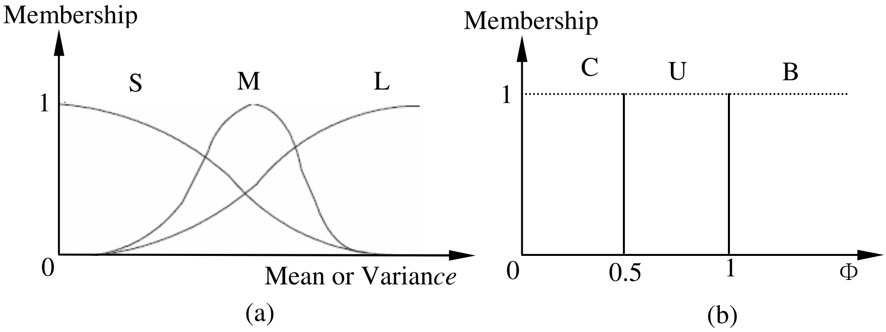


Fig. 2. (a) Input member functions. (b) Output member functions.

4.2 TCP Vegas_M

In this subsection we present our modification to TCP Vegas congestion avoidance mechanism, while slow start and congestion recovery algorithms of Vegas_M are the same as that of TCP Vegas [5],

According to the analysis of TCP Vegas congestion control mechanism, the reliability, the variation range of *RTT* control throughput and delay of TCP Vegas. However, according to [6], it is inappropriate for TCP Vegas congestion control mechanism to be applied to MANET without modification. That is mainly because the mechanism still don't take the bit error rate (BER) of wireless channel into account and the undesirable response caused by the BER drastically decreases the sending rate, resulting in an underutilization.

Based on the FEDM algorithm, we improve on TCP Vegas congestion avoidance mechanism over the MANET. Our main aim of this algorithm lies in: （1）it will keep TCP Vegas congestion control mechanism so as to makes the algorithm scalable to various networks; （2）when the wireless channel performance become bad and lead to BER increasing in wireless links, TCP Vegas sender could adjust data transmission speed and ensure its reliability; （3）the algorithm could improve on TCP throughput, delay and networks resources utilization. We propose a modified congestion avoidance mechanism as follows:

```
Algorithm 1: (congestion)
$if\ diff < \alpha$ {
$cwnd = cwnd + 1$ };
$else\ if\ \alpha < diff < \beta$ {
```

```
$cwnd = cwnd$ };
$else\ if\ diff > \beta$ {
$cwnd = cwnd - 1$ };$end\ if$ ;
```

Algorithm 2: (*uncertain*)

```
$if\ diff < \alpha$ {
$cwnd = cwnd$ };
$else\ if\ diff > \alpha$ {
$cwnd = cwnd - 1$ }; $end\ if$ ;
```

Algorithm 3: (*bit error*)

```
$if\ diff < \beta$ {
$cwnd = cwnd + 1$ }
$else\ if\ diff > \beta$ {
$cwnd = cwnd$ }; $end\ if$ ;
```

When the FEDM output value is congestion, this is an indication that the network is fully saturated the congestion, then avoidance mechanism implement algorithm 1, which is same as [5]. When the FEDM output value is uncertain, it is indication that network state is congestion or channel deteriorative, so TCP should decline actual throughput to rapidly relieve network congestion state or decline number of error packet, decrease number of retransmission of error packets. Then, avoidance mechanism implement algorithm 2. When the FEDM output value is bit error, this is indication that the network is not fully saturated and that network has enough bandwidth resource. TCP sender does not rapidly decline congestion window, only when channel lies in very bad state. So TCP avoidance mechanism implement algorithm 3.

5 Simulation Results

In order to validate the TCP Vegas_M algorithm, we use the well-known ns simulator [16] from UCB/LBNL. This paper does not address the link failure problem, which is caused by nodes mobility. So, we consider a string and static topology with 10 nodes as show in figure 3, the interval between any neighbor nodes is 250 meters, and bandwidth is 2 Mbps. To demonstrate the superiority of TCP Vegas_M, we compare its performance to two traditional TCP congestion control algorithm in wireless networks, TCP Reno, TCP Vegas [4]. We run each simulation for 200 seconds to get the average throughput with varying number of hops between TCP sender nodes and TCP receiver nodes.

Fig. 3. Simulation String Topology

First, to compare the throughput performance with different TCP algorithms under severe channel condition, packet losses probability is 1%. So, we set up a single TCP connection between TCP sender nodes and receiver nodes. TCP maximum congestion window is 32, packet size is 512 bits.

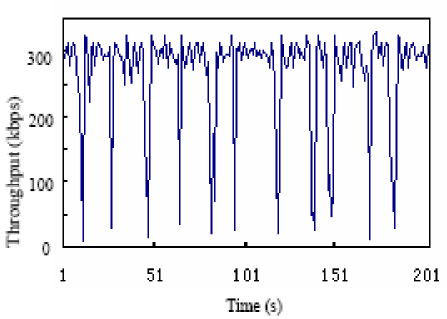


Fig. 4. TCP Reno throughput

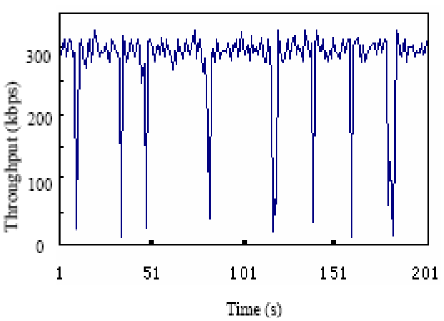


Fig. 5. TCP Vegas throughput

Figure 4,5,6 show simulation results of throughput of different TCP algorithms. As result, the throughput of TCP Reno and TCP Vegas is high frequency oscillation, this is because that TCP Reno and Vegas didn't distinguish losses packet between network congestion and channel error packet, every time after losses packet, TCP rapid degrade it congestion window, and lead to TCP performance degraded seriously. Figure 6 shows that packet losses only cause slight instability to the TCP Vegas_M congestion window, and it achieves higher throughput than former TCP under the same wireless channel conditions, because TCP Vegas_M can distinguish different losses packet.

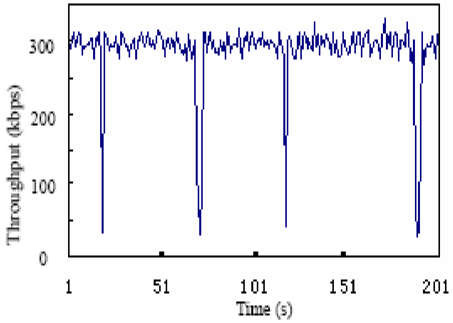


Fig. 6. TCP Vegas_M throughput

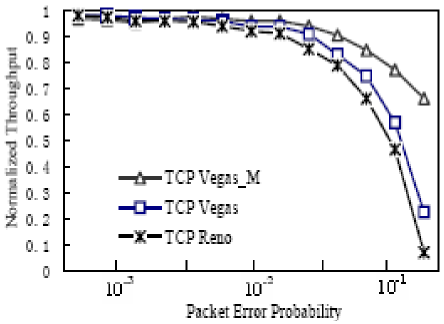


Fig. 7. 1 Hop Normalized throughput

Next, we investigate the impact of network congestion and channel error on three TCP algorithms under the different hops. The value of maximum congestion window is same as former, packet size is 1460 bytes. The normalized throughput results show that three TCP algorithm match closely for small packet error probability. However, with higher packet error probability, the performance of TCP Reno and Vegas degrade very rapidly, it indicate that two algorithms don't distinguish network congestion from channel error, every packer loss triggers TCP cut its congestion window, and lead to TCP throughput be degraded. Our algorithm simulation show that packet losses by packet error only make for small disturbances to TCP sending window, and packet error losses not result in drastic decline of TCP performance.

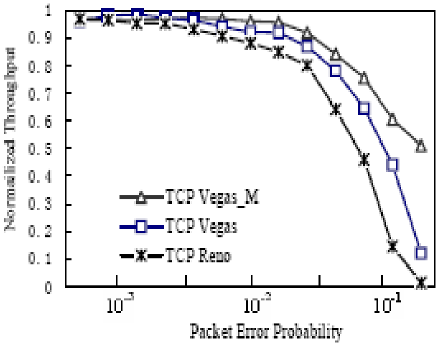


Fig. 8. 5 Hop Normalized Throughput

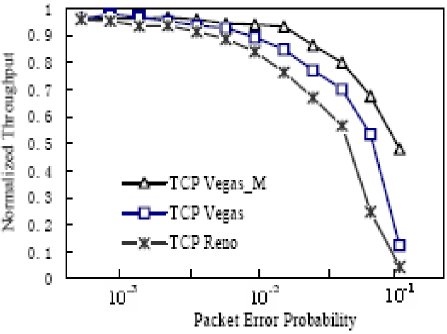


Fig. 9. 10 Hops Normalized Throughput

6 Conclusions

In this paper, we have presented novel TCP congestion control algorithm (TCP Vegas_M) in the mobile ad hoc networks. This TCP algorithm depended on FEDM fuzzy logic theory to distinguish network congestion states and wireless channel states, and based on ad hoc networks states (congestion, uncertain, and bit error), TCP sender adopt different congestion avoidance algorithms to control congestion window size, aims is to optimize ad hoc network throughput and achieve better performance. Simulation Results show that the algorithm can be easily implemented with fewer overheads and can effectively improve TCP sending nodes throughput, and enhance network utilization.

This paper has not considered TCP Vegas slow-start and recovery mechanism. At the same time, we mainly simulated algorithm with string and static topology scenarios, and more related work will be conducted in future work.

Acknowledgments. This work is partly support by the National Defense Foundation of China， Natural Science Foundation Project of CQ CSTC, 2006BA6016, and Chongqing University Postgraduates' Science and Innovation Fund: 200701Y1A0030189.

References

1. Li, X., Chua, K., Kong, P., Jiang, S.: The impact of lossy links on tcp performance in IEEE802.11 based ad hoc Networks. In: WCNC 2005/IEEE Communications Society, pp. 1545–1550 (2005)

2. Srijith, K.N.: Improving the performance of TCP Vegas and TCP SACK: investigations and solutions, www.srijith.net/publications/Srijith_MSc_Thesis.pdf
3. Fu, Z., Meng, X., Lu, S.: How Bad TCP Can Perform In Mobile Ad Hoc Networks. In: ISCC 2002. Proceedings of the Seventh International Symposium on Computers and Communications (2002)
4. de Oliveira, R., Braun, T.: A delay-based approach using fuzzy logic to improve TCP error detection in ad hoc networks. In: WCNC 2004/IEEE Communications Society, pp. 1666–1671 (2004)
5. Brakmo, L.S., Peterson, L.L.: TCP Vegas: end to end congestion avoidance on a global. Internet. IEEE Journal on Selected Areas in Communications 13(8), 1465–1480 (1995)
6. Ahn, J., Danzig, P., Liu, Z., Yan, L.: Evaluation of TCP Vegas: emulation and experimen. Computer Communication Review 25(4), 185–195 (1995)
7. La, R.J., Walrand, J., Anantharam, V. : Issues in TCP Vegas,
www.eecs.berkeley.edu/~ananth/ 1999-2001/Richard/IssuesInTCPVegas.pdf
8. Cheng, L., Marsic, I.: Fuzzy Reasoning for Wireless Awareness. International Journal of Wireless Information Networks 8(1), 15–25 (2001)
9. Cheng, L., Marsic, I.: Fuzzy Reasoning for Wireless Awareness. International Journal of Wireless Information Networks 8(1), 15–26 (2001)
10. El-Sayed, H.M.: Performance evaluation of TCP in mobile ad hoc networks. In: IIT 2005. The second International Conference on Innovations in Information Techonnlogy (2005)
11. Chandran, K., Raghunathan, S., Venkatesan, S., Prakash, R.: A Feedback Based Scheme For Improving TCP Performance In Ad-Hoc Wireless Networks. In: ICDCS 1998. Computer International Conference on Distributed Computing Systems, Amsterdam, Netherlands (May 1998)
12. Liu, J., Singh, S.: ATCP: TCP for mobile ad hoc networks. IEEE JASC 19(7), 1300–1315 (2001)
13. Fu, Z., Greenstein, B., Meng, X., Lu, S.: Design and Implementation of a TCP-Friendly Transport Protocol for Ad Hoc Wireless Networks. In: Proceedings of IEEE ICNP 2002, Paris (2002)
14. Chen, K., Xue, Y., Nahrstedt, K.: On Setting TCP's Congestion Window Limit in Mobile Ad Hoc Networks. In: ICC 2003. Proc. IEEE Intl. Conf. on Communications (2003)
15. Srijith, K.N., Jacob, L., Ananda, A.L.: TCP Vegas-A:Solving the Fairness and Rerouting Issues of TCP Vegas. In: IPCCC. C Proceedings of 22nd IEEE International Performance, Computing, and Communications Conference, Phoenix, Arizona, pp. 309–316 (April 9 - 11, 2003)
16. NS Simulator, http://www.isi.edu/nsnam/ns

A Study on the Binary Exponential Backoff in Noisy and Heterogeneous Environment

Khoder Shamy, Chadi Assi, and Lei Guang

Concordia University
Montréal, Québec, Canada, H3G 1M8
{k_shamy,assi,l_guang}@ece.concordia.ca

Abstract. Recently, some proposals have suggested that maintaining the same contention window (CW), or reducing it, for nodes suffering packet losses, due to channel transmission impairments, is effective in increasing the performance of the IEEE 802.11 in noisy environment. Our study presented inhere will prove analytically and via simulations that this should not be necessarily the case. To facilitate our analysis, we consider two binary exponential backoff (BEB) algorithms in our study: a standard BEB where a host increases its CW upon every packet loss (collision or transmission error) and another access method with a capability to differentiate between the type of losses; here, a host experiencing a loss will increase its CW only after a collision and remain in the same backoff stage otherwise. We show that the second access procedure outperforms the standard BEB when the network is lightly loaded. However, in a congested network, this quick recovery property results in intensifying the collisions among contending nodes and hence yields a poor system performance. We propose a hybrid method that takes advantage of both access methods to achieve better throughput under various network loads.

1 Introduction

The IEEE 802.11 protocol has become the predominant technology for wireless local area networks (WLAN). One of the most important elements of the 802.11 is its medium-access control (MAC); the MAC protocol is used to provide arbitrated access to a shared medium, in which several terminals access and compete for using the network resources. The IEEE 802.11 wireless networks employ the distributed coordination function (DCF) as the primary access mechanism; it is based on the carrier sense multiple access with collision avoidance (CSMA/CA) protocol and binary exponential backoff. The performance of an IEEE 802.11 network largely depends on the operation of this backoff mechanism. Accordingly, there have been several research efforts for analyzing the saturation throughput achieved under DCF in single hop WLANs. The binary exponential backoff (BEB) of DCF is used for resolving collisions among terminals attempting to simultaneously transmit on the channel. To ensure packet transmission reliability, MAC acknowledgement frames (ACK) are used to indicate the correct reception

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 397–408, 2007.

of the data packets. When an ACK frame is not received upon a transmission, the sender assumes its packet has been lost due to collision and accordingly invokes the BEB mechanism for retransmission. However, packet losses may also occur from other transmission errors, which may arise from the combination of both path loss and receiver noise. Previous research efforts have evaluated Wireless LANs access methods in the presence of transmission errors. For example, the authors of [1] evaluated, through simulations, the performance (throughput and fairness) of different access methods by varying the bit error rate (BER) of the channel. The authors of [2] presented analytical studies of the saturation throughput of the IEEE 802.11 access method; here a known and fixed BER is assumed in [2] or a BER derived from the bit-energy-to noise ratio as perceived by the receiver is used to determine the packet error rate (PER) which is needed for the analytical model to derive expressions for the throughput. However, deriving the PER from the BER does not seem to be a feasible and practical method since this information is not readily available at the transmitter. The authors of [9] noted that when a packet is lost due to transmission error, doubling CW is inappropriate and could seriously degrade the performance. The authors of [5] studied the performance of the standard BEB in noisy environments and showed that it results in poor performance since it does not have any mean to differentiate between packet losses. They proposed an enhancement for BEB where a node treats any failed transmission due to channel corruption as successful and schedule the retransmission accordingly. They presented an estimation method for differentiating between the causes of packet losses; however, their estimation method depends on measuring the number of active stations and their obtained results show a strong variation of the estimated channel error rate from the actual rate. In this paper we compare the performance of two access methods (section 2) in the presence of channel errors in order to study the relationship between the operation of CW and the system throughput in noisy and heterogenous environment. We show that a simple access method (i.e., BEB_1) without any capability for differentiating between packet losses yields poor performance when the load on the network is light, as opposed to a second access method (BEB_2) where a node performs backoff only if the packet is lost due to collision and otherwise remain in the same backoff stage. However, when the load on the channel is high, the first access method achieves higher overall utilization. We also compare the two access methods when different stations experience different packet error rates. We present (section 3) an online extension for an effective PER estimation tool initially proposed by [3] and show its effectiveness; then we propose a hybrid approach that takes advantage of both access methods to achieve better system throughput under various network loads. We conclude our work in section 5.

2 Analysis of 802.11 in Noisy Environment

The model we use in our study for BEB is based on Bianchi's model [10]. Denote p_c as the conditional collision probability which is constant and independent of the number of retransmissions incurred, let p_e represent the PER and N is the total number stations. A station is assumed to always have packets to send. Upon detecting a packet loss, a station decides whether the packet was lost due to collision or transmission error. If the station infers collision, then the standard BEB mechanism is followed and CW is doubled. If the packet is lost due to noise, then the station could either choose to double its backoff and schedule a retransmission or remain in the same backoff stage or treat the transmission as successful and accordingly resets CW and retries the transmission [5]. Denote by τ, the transmission probability of a station in a randomly chosen time slot:

$$\tau = \sum_{i=0}^{m} b_{i,0} = \frac{2(1-2p_f)}{(1-2p_f)(W_0+1) + p_f W_0 (1-(2p_f)^m)} \tag{1}$$

$$where \quad p_f = 1-(1-p_c)(1-p_e) \quad \& \quad p_c = 1-(1-\tau)^{N-1} \tag{2}$$

However, if a station uses the same backoff stage upon an erroneous transmission, then:

$$\tau = \sum_{i=0}^{m} b_{i,0} = \frac{2(1-2\alpha)}{(1-2\alpha)(W_0+1) + \alpha W_0 (1-(2\alpha)^m)} \tag{3}$$

where, $\alpha = \frac{p_c}{p_c+p_s}$ and p_s ($p_s = 1 - p_f$) is the success probability. Let S be the normalized system throughput, defined as the fraction of time the channel is used to successfully transmit the payload bits. Let P_{tr} be the probability that there is at least one transmission in the considered slot time and the successful probability P_t is the probability that exactly one station transmits, hence:

$$P_{tr} = 1-(1-\tau)^N \quad and \quad P_t = \frac{N\tau(1-\tau)^{N-1}}{P_{tr}} \tag{4}$$

We define the saturation throughput of the network as:

$$S = \frac{(1-p_e)P_s s_d}{P_i\sigma + (1-p_e)P_s T_s + P_c T_c + p_e P_s T_f} \tag{5}$$

where $P_s = P_{tr}P_t$ is the probability of a successful transmission, $P_i = 1 - P_{tr}$ is the probability of an idle slot, $P_c = 1 - P_i - P_s$ is the collision probability, s_d is the average packet payload size, and σ is the duration of an idle slot time. T_s, T_c and T_f are the average time the channel is sensed busy.

2.1 Analysis - Part I

In this section we present numerical results from the analytical models in order to better understand the behavior of the two access methods presented earlier and expose the salient differences between them. We study the effect of packet error rates on the system performance; we consider the basic access mode and we assume here all the stations experience the same p_e. All the parameters used in our study follow those of DSSS; we consider two channel rates, 1Mbps and 11Mbps, and data payload size of 1050 bytes. At 1Mbps, BEB_1 outperforms BEB_2 (see figure 1(a)), except when the number of stations is very low (e.g., below 3 stations when the PER is 0.6). These results suggest that it is advantageous to follow the standard operation of BEB upon a failed transmission regardless of the reason the transmission fails. This is mainly due to the fact that when the channel rate is low, a small number of stations saturate the channel; therefore, trying to quickly recover from transmission errors by not backing off (as in BEB_2) intensifies the contention on the channel and results in less air time that can be efficiently utilized. Alternatively, when the channel rate is higher (e.g., 11Mbps), BEB_2 access method outperforms BEB_1 when the channel is lightly loaded (Figure 1(b)). For example, when p_e is 0.6, it is advantageous to follow BEB_2 as long as the number of contending stations is less than 15. That is, when operating with unsaturated channel and in the presence of channel noise, it is advantageous for stations to quickly recover from packet losses, by retransmitting from the same backoff stage, since unnecessary backing off when resources are available will only leave much of the air time unused and result in poor overall performance. However, as the number of stations increases, maintaining the same backoff stage upon a failed transmission (not collision) would exacerbate the contention and hence intensify collisions among contending stations. In order to further understand these findings, we consider the average time needed to successfully transmit a packet; the authors of [11] have shown that the average time $E[T]$ for servicing a packet in an N user WLAN can be written as; $E[T] = K + \frac{E[B]}{N} + \frac{E[C]}{N_c}$, where K represents the sum of the time incurred in sending PHY and MAC layers' headers, the payload and the time spent when the system incurs a failed transmission due to channel noise. $E[B]$ is the average time a station spends in backoff, $E[C]$ is the average time a station spends in collision and N_c is the average number of stations involved in a collision:

$$E[B] = \frac{\sigma}{\tau(1-p_c)} \quad \mathit{and}, \quad E[C] = \frac{p_c}{1-p_c} T_c \tag{6}$$

$$N_c = \frac{N\tau - N\tau(1-\tau)^{N-1}}{1-(1-\tau)^N - N\tau(1-\tau)^{N-1}} \tag{7}$$

Clearly, the term $\frac{E[B]}{N} + \frac{E[C]}{N_c}$ in the expression of $E[T]$ is the only term affected by the choice of the access method. Therefore, the lower this term, the lower is the time it takes to service successfully a packet and hence the higher is the overall achieved throughput. In Figure 1(c), we show the backoff time normalized by N and the collision time normalized by N_c for both access methods in the case where the channel error rate is 0.6. We see that stations under BEB_2 constantly have lower backoff delays; this is because stations suffering packet losses always attempt to recover from these losses by retransmitting from their current backoff stage. Alternatively, stations under BEB_1 experiences much larger backoff delays, especially when the number of stations is small. This is mainly attributed to the fact that packets that are lost due to channel error will force stations to continuously backoff, although contention on the channel is not the problem.

On the other hand, the average time spent in collisions under BEB_2 is larger; this is due to the higher frequency of access on the channel by stations following BEB_2, which results in intensifying the collision. Indeed, it is the reduction in time spent in backoff (when the number of stations is small) that is allowing BEB_2 to achieve higher throughout. Here, although the time spent in collision is higher, the quick retransmission attempt by stations to recover from packet loss will yield an improvement in the system performance, especially when the channel is lightly loaded. This is shown further in Figure 1(d) where we look at the sum of both times; when the load is light (e.g., $N \leq 15$), the total time is smaller than that of BEB_1 which explains the higher throughput achieved in Figure 1(b). Alternatively, when the load is higher, the figure shows it is not advantageous to resort to quick recovery from packet loss as this will only exacerbate collisions on the channel and result in lower overall system performance. Now, in order to maximize the system throughput, the authors of [12]suggested a method to optimize the performance of 802.11 by turning off the BEB access method and derived the optimal initial CW, CW^*_{min} (from τ^*, which is the solution of $(1-\tau^*)^N = \frac{T_c^*+1}{T_c^*}(1-N\tau^*)$) that all stations should utilize. It is clear however that τ^* is strongly dependent on the number of stations, N, which is not trivial to estimate. We show in Figure 1(b) the throughput obtained by this access method; clearly, when the error rate is small, the performance is significantly improved; here, in this method, all stations equally attempt to access the channel so that they achieve the maximum possible throughput.Therefore, a station does not need to differentiate between the causes of packet losses. As the error rate increases, the optimized access method performs similar to BEB_2 when the load is light and similar to BEB_1 when N is higher.

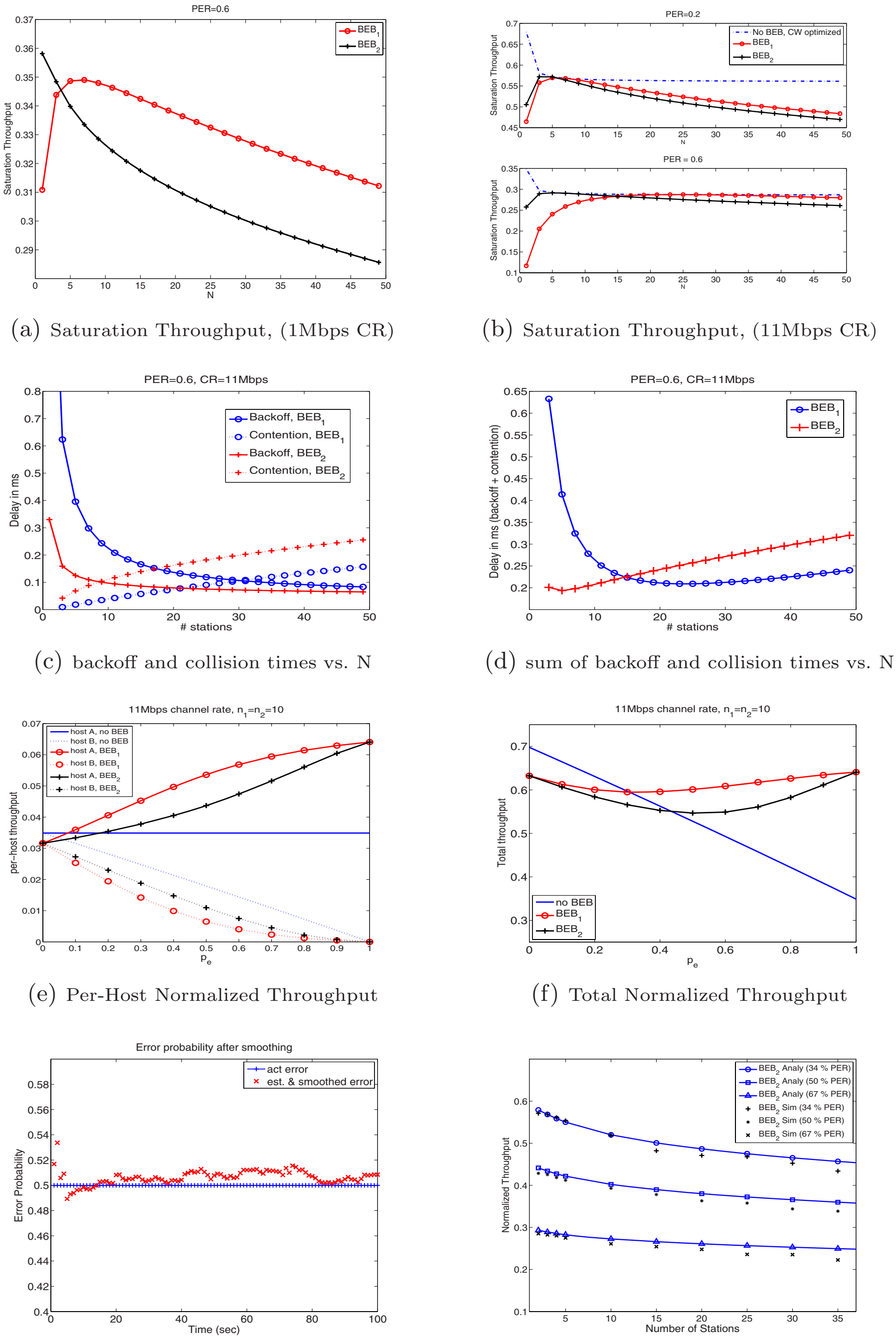
2.2 Analysis - Part II

We extend our analysis to consider a network with heterogenous conditions wherein hosts (labeled $l = 1, ..., N$) experience different transmission errors. For simplicity, we consider two classes of nodes, where all stations (n_1) within one class (C_1) do not suffer any channel impairments and the remaining stations (C_2, $n_2 = N - n_1$) experience all the same transmission error, p_e. Here, under BEB_1, the transmission probability ($\tau^{(1)}_{l,i}$, $i = 1, 2$) for a station l in class i is given by (1) and its transition probability ($p^{(1)}_{f,l}$) is:

$$p^{(1)}_{f,l} = 1 - (1 - \tau^{(1)}_1)^{n_1 - 1}(1 - \tau^{(1)}_2)^{n_2} \qquad and, \tag{8}$$

$$p^{(1)}_{f,l} = 1 - (1 - p_e)(1 - \tau^{(1)}_1)^{n_1}(1 - \tau^{(1)}_2)^{n_2 - 1} \quad (if\ l \in C_1) \tag{9}$$

With N stations, equations (1),(8), and (9) provide a set of non linear equations that can be solved numerically for $\tau^{(1)}_1$, $\tau^{(2)}_1$, and $p^{(1)}_{f,l}$ ($l \in C_1$ or C_2). Clearly, hosts belonging to different classes will access the channel with different



(a) Saturation Throughput, (1Mbps CR)

(b) Saturation Throughput, (11Mbps CR)

(c) backoff and collision times vs. N

(d) sum of backoff and collision times vs. N

(e) Per-Host Normalized Throughput

(f) Total Normalized Throughput

(g) Smoothing p_e

(h) Validation of Analytical Model

Fig. 1. Analytical and Simulation Figures

transmission probability, as determined by the access method. The probability that a host l successfully transmits is:

$$p_s^l = \begin{cases} \tau_1^{(1)}(1-\tau_1^{(1)})^{n_1-1}(1-\tau_2^{(1)})^{n_2} & \text{if} \quad l \in C_1 \\ \tau_2^{(1)}(1-\tau_1^{(1)})^{n_1}(1-\tau_2^{(1)})^{n_2-1} & \text{if} \quad l \in C_2 \end{cases} \tag{10}$$

The sum of the probabilities of successful transmissions for all stations is $P_s = \sum_{l=1}^{N} p_s^l$ and the probability of at least one station attempting transmission is:

$$P_{tr} = 1 - P_i = 1 - (1-\tau_1^{(1)})^{n_1}(1-\tau_2^{(1)})^{n_2} \tag{11}$$

where P_i is the probability that the medium is idle. Hence, the collision probability is $P_c = 1 - P_i - P_s$. Finally, one can determine the per-host (l) throughput:

$$S_l = \begin{cases} \dfrac{p_s^l s_d}{P_i\sigma + P_s T_s + P_c T_c} & \text{if} \quad l \in C_1 \\ \\ \frac{(1-p_e)p_s^l s_d}{P_i\sigma+(1-p_e)P_s T_s+P_c T_c+p_e P_s T_f} & \text{if} \quad l \in C_2 \end{cases} \tag{12}$$

and the normalized throughput for the system is $S = \sum_{l=1}^{N} S_l$. In a similar way, one can determine the individual throughput for a host using the second access method. We will compare the per-host throughput under the two access methods to that achieved in a network where BEB is absent [12]. For that case, all nodes are forced to use the same access probability $\tau_l = \tau^*, l = 1...N$ and the throughput is computed using (12). Figure 1(e) shows the per-station throughput for two nodes (A, a station with ideal channel conditions, and B, a station with bad channel quality) when varying the error transmission rate in a network with 20 nodes, 10 of which experience transmission errors. Under BEB_1 , the throughput for host B decreases quickly as p_e increases while the throughput for host A increases; this is because in this scenario, 50% of the hosts are experiencing transmission errors and hence are forced to continuously backoff upon every packet loss thereby leaving the channel idle for other hosts (50%) to collectively access with less contentions. On the other hand, under BEB_2, host B (A) obtains a higher (lower) throughput than it would under BEB_1. Here, a host experiencing transmission impairments enjoys higher access to the channel in order to overcome the channel error and therefore achieve higher throughput as opposed to a host experiencing the same error under BEB_1. These results demonstrate that although BEB_2 may not achieve higher overall system throughput than BEB_1, especially when the number of nodes is high (as shown in Figure 1(f)), BEB_2 enables stations experiencing transmission errors to obtain a better access to the channel and hence improve their throughput. Finally, it is to be noted that BEB_2 improves slightly the fairness among stations as opposed to BEB_1 where some hosts obtain large throughput at the expense of starving other hosts. Figure 1(e) shows also the per-host throughput achieved by an optimal access method where BEB is absent [12]. The figure shows that when there is no error (or small error), this system achieves better individual and overall system throughput.

However, as the channel condition deteriorates, host A consistently receives the same access to the channel and hence attains the same throughput while host B, receives a decreasing throughput as p_e increases. Note, however, that the throughput achieved by host B under this access method is always higher than that obtained under the other access methods due to the fact that the access probability is derived to achieve maximum throughput. This result also shows that this access method yields the best fairness among different hosts. With respect to the overall system throughput, Figure 1(f) shows that the optimized access method achieves the highest throughput when the error rate for stations in C_2 is small. However, when the channel conditions deteriorate, the optimal access method severely degrades the network performance since hosts are not able to *dynamically* adjust their access to efficiently utilize the channel. However, the other two access methods allow stations not experiencing any channel error to increase their access and hence achieve higher overall throughput. Note that BEB_1 is a more greedy access method than BEB_2, since stations experiencing transmission impairments are forced to unnecessarily backoff, thereby allowing other stations better access the channel, and accordingly yielding a higher overall throughput.

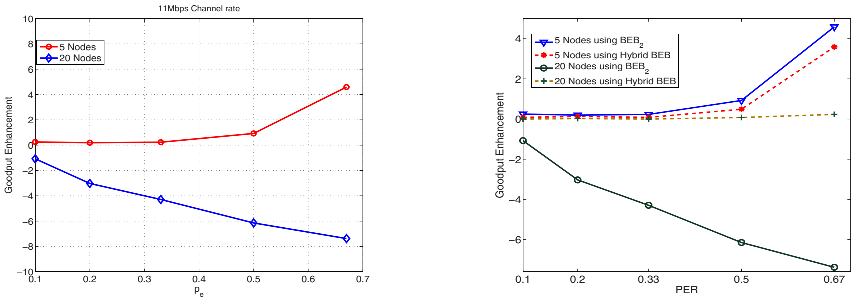
3 Error Estimation and Performance Analysis

Many studies have shown that WLANs exhibit time varying characteristics; namely, the quality of received signals changes dramatically even over short time intervals due to multiple causes (noise, attenuation, interference, multipath propagation, and host mobility) [4]. Estimation of channel quality has been a challenging issue for researchers; currently most work presented in the literature, either focus on the PHY layer through SNR values [8] which are inaccurate or suggests modification of the standard to add a NAK frame [9]. Recently, the authors of [3] presented a method for estimating the channel quality at the MAC layer. The method is based on few observations about the operation of CSMA/CA method. To specify the admissible transmission times, the authors defined four events: *idle slots*, *other transmissions*, *successful transmissions* and *unsuccessful transmissions* [3]. The probability of a collision by one node is then precisely the probability that at a slot boundary the channel is busy due to a transmission by one or more other stations. If over some time period, a station transmits T times and of these A are successful because an ACK is received and there are also R slots in which the station does not transmit and that I of these are idle. The collision and error probabilities can be estimated [3]:

$$p_c = \frac{R-I}{R} = \frac{\#\text{other transmits}}{\#\text{idle} + \#\text{other transmits}} \quad and, \quad p_e = 1 - \frac{1-\frac{T-A}{T}}{1-p_c} \tag{13}$$

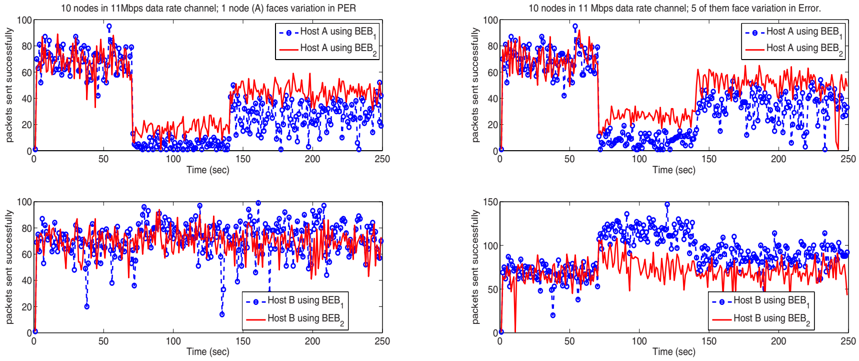
providing that $0 \leq p_e \leq 1$, p_e is determined from successful probability expression $p_s = (1-p_c)(1-p_e)$. We found that this estimation method was the most efficient and accurate, and does not require any changes of the current standard. We note here that the authors of [3] have provided an off-line validation of the estimator. The key issue is to keep track of both busy slots (*other transmissions*) as well as the idle slots during a predetermined time interval of our choice from which p_c and p_e are estimated. An extension we made to the online estimator was to use the Auto Regressive (AR) filter in order to reduce the variations of the estimated error as; $\hat{p}_e(n) = \delta \times \tilde{p}_e(n) + [1-\delta] \times p_e(n-1)$ where $\hat{p}_e$ is the AR smoothing of the error probability and $\tilde{p}_e$ is the measured error probability for an individual node taken at the n^{th} time interval and δ is the correlation coefficient ($\delta = 0.125$ was used throughout the experiments) and $\tilde{p}_e$ is the measured error probability. Figure 1(g) shows the estimated $\hat{p}_e$ vs. time when the channel rate is 11Mbps and the actual packet error rate is fixed at 50%. The estimated $\hat{p}_e$ is sampled every 1-second; we can clearly see now the remarkable performance of the online estimator where a variation of about 1% between the actual and the estimated error rate is observed.

Next, we validate the analytical model presented earlier with results obtained from simulations using *QualNet* simulator [7]. The IEEE 802.11b was used throughout the experiments. We assume that all stations experience the same packet corruption rate and that a station has always a packet to transmit (of 1050 bytes payload size) using the basic access mode. The channel rate is fixed at 1Mbps. We obtain results (normalized system throughput) under different packet error rates and the results are presented in Figure 1(h) . The figure shows that the analytical results are comparable to those of the simulation (both results are for BEB_2) as the load increases for various channel conditions (i.e., different p_e). Next, we compare the performance of the two access methods BEB_1 and BEB_2. We use the throughput enhancement [5] as a metric for our comparison: $\nabla S = \frac{S'-S}{S} \times 100$ where S' and S are the system throughput achieved under BEB_2 and BEB_1 respectively. Figure 2(a) shows the throughput enhancement obtained from simulation results for two networks of 5 and 20 nodes while varying channel error rates. Note that at any given time, all nodes in the network experience the same transmission impairments. Clearly, the figure shows that when the number of nodes is low ($N = 5$), BEB_2 outperforms BEB_1 under various channel conditions. However, when the number of hosts increases ($N = 20$), the throughput enhancement decreases (below zero) and BEB_1 outperforms the second access method. This has also been shown from the analytical results in Figure 1(b). The conclusion from these results is that it is not always good to recover from channel noise by performing quick retransmission because the channel may be congested and fast retransmission would only exacerbate the collision. Our result counters the claim of [5] where the authors designed a smart access method and concluded that it is always advantageous to reset the CW upon an unsuccessful transmission. The authors of [12] measured the throughput penalty obtained when default BEB (N=10, 11Mbps channel rate) is operating under the presence of transmission errors. They concluded that this throughput penalty is attained because BEB is not capable of differentiating between packet losses. Accordingly, our results showed the same (i.e., for N=10); we have also shown that even with the presence of loss differentiation, the throughput penalty will be higher when the number of nodes accessing the medium is large. Now,



(a) BEB_1 and BEB_2 *Goodput* (b) BEB_2 and Hybrid BEB Goodput.

Fig. 2. Goodput results for BEB_1, BEB_2 and Hybrid BEB. (11 Mbps CR)



(a) Only 1 node suffers noise (host A), the other 9 nodes have error-free channel conditions (host B one of them).

(b) 5 nodes suffer noise (host A one of them), the other 5 compete in error-free channel (host B one of them).

Fig. 3. Throughput attained as PER is varied (0%, 67%, & 34%) discriminatively

we consider a network with 10 competing stations where only one node suffers variation in transmission errors (Fig. 3(a)) and another scenario of 10 competing nodes where 5 of them face transmission errors (Fig. 3(b)). Figure 3(a) shows the throughput acquired by host A (experiencing errors) and host B (no error); host A receives a lower throughput under BEB_1 access method, especially at higher channel error rates. However, the host throughput improves under the second access method (BEB_2) due to the faster retransmissions upon packet loss due to channel error. Note that host B will receive the same throughput under the two access methods. This is due to the fact that although BEB_1 penalizes host A, the gain in channel access is shared among all competing hosts (9 in this case) and accordingly, the gain per host is relatively minor. Alternatively, when the number of stations experiencing transmission errors is higher (e.g., 5 in Fig. 3(b)), host A obtains much better throughput under BEB_2 than that obtained under BEB_1. When the PER increases, the throughput of host A is further impacted and host B will gain more access to the channel, since 5 nodes (out of the 10) are suffering transmission errors which leads to a significant increase in channel idle time and hence, host B will have better chance for successful transmissions especially as the PER increases. This would actually seem like the channel is operating with less number of nodes. The gain in throughput is attributed to the reduction in access from hosts experiencing transmission errors (both under BEB_2 and even more under BEB_1). The figure shows that host B obtains slightly higher throughput under the first access method than the second one. Overall, the results reveal that BEB_2 yields better per host throughput and hence better long term fairness as opposed to BEB_1 where hosts with transmission errors will be severely penalized although the overall system performance of BEB_1 may be slightly better than that under BEB_2. These results have been corroborated from our analysis of the analytical model.

4 An Adaptive Access Method

Clearly, it has been shown that both BEB_1 and BEB_2 offer some advantages depending on the load of the network. We will attempt to derive an enhanced access method to combine the advantages of the two backoff procedures. First, we note that the authors of [6] derived an expression that enabled them to determine when the channel is used in an optimized manner based on the transmission attempt probability. Accordingly, they proposed a novel method whereby hosts are forced to compete with the same CW whose value depends only on the network load and is derived after measuring the mean number of idle slots between two transmission attempts and comparing it with the optimal one $\bar{n}_{i\infty}^{opt}$ that is a computable constant; (it is equal to 5.68 for the 802.11b parameters), the complete derivation of $\bar{n}_{i\infty}^{opt}$ can be found in [6]. Our proposed hybrid method benefits from that approach in probing the load on the channel. Hence, we keep track of the mean number of idle slots between two transmission attempts, and when the channel is highly loaded (measured $\bar{n}_i < \bar{n}_{i\infty}^{opt}$) we turn on BEB_1 mechanism upon a packet loss, while BEB_2 is used if the channel is seen lightly loaded (i.e. $\bar{n}_i < \bar{n}_{i\infty}^{opt}$). Figure 2(b) plots the throughput enhancement presented in the previous section for 5-nodes and 20-nodes network when implementing BEB_2 and the hybrid BEB while varying the transmission corruption rate. Both methods achieve almost the same enhancement when the number of nodes is low. However, it is clear that the performance of BEB_2 deteriorates when the number of nodes is 20 (negative enhancement), while the hybrid BEB performs the same as BEB_1 (zero enhancement). We conclude that using the hybrid adaptive approach would take advantage of both BEB_1 and BEB_2 to achieve better system throughput under various network conditions.

5 Conclusion

We investigated the performance of two Wireless LAN access methods in the presence of channel transmission impairments; a standard that increases the contention window after every packet loss and another access method with a capability of differentiating between transmission errors and collision. Hosts implementing the second access method will increase their CW only after collision, reset if the transmission is successful and maintain the same backoff stage otherwise. This enables hosts experiencing channel impairments to quickly recover from transmission errors. Our results revealed that the second access method outperforms the standard BEB when the network load is light; however, as the load increases, the quick recovery property of the second access method intensifies collisions among contending hosts especially at higher PER. We presented an adaptive approach that integrates the operations of both access methods by deploying some network probing scheme to measure the load on the network and accordingly adapt the access of the nodes.

References

1. Lopez-Aguilera, E., Heusse, M., Rousseau, F., Duda, A., Casademont, J.: Evaluating Wireless LAN Access Methods in Presence of Transmission Errors. In: IEEE INFOCOM (2006)
2. Chatzimisios, P., Boucavalas, A.C., Vistas, V.: Performance Analysis of IEEE 802.11 DCF in Presence of Transmission Errors. In: IEEE ICC (June 2004)
3. Malone, D., Clifford, P., Leith, D.J.: MAC Layer Channel Quality Measurement in 802.11. IEEE Comm. Letters 11(2) (February 2007)
4. Aguayo, D., Bicket, J., Biswa, S., Judd, G., Morris, R.: Link Level Measurements from an 802.11b Mesh Network. In: Proc. of ACM SIGCOMM, Boston, USA (2004)
5. Nadeem, T., Agrawala, A.: IEEE 802.11 DCF Enhancements for Noisy Environments. In: Proc. of IEEE PIMRC (September 2004)
6. Heusse, M., Rousseau, F., Guillier, R., Duda, A.: Idle Sense: An Optimal Access Method for High Throughput and Fairness in Rate Diverse Wireless LANs. In: Proc. of ACM SIGCOMM (August 2005)
7. SNT, QualNet Simlator, http://www.scalable-networks.com/
8. Qiao, D., Choi, S.: Goodput Enhancement of IEEE 802.11a Wireless LAN via Link Adaptation. In: Proc. IEEE ICC, Finland (2001)
9. Pang, Q., Liew, S.C., Leung, V.C.M.: Design of an Effective Loss-Distinguishable MAC Protocol for 802.11 WLAN. IEEE Comm. Letters 9(9) (September 2005)
10. Bianchi, G.: Performance Analysis of the IEEE 802.11 Distributed Coordination Function. IEEE JSAC 18(3), 535–547 (2000)
11. Medepalli, K., Tobagi, F.A.: Throughput Analysis of IEEE 802.11 Wireless LANs using an Average Cycle Time Approach. In: Proc. of GLOBECOM (2005)
12. Medepalli, K., Tobagi, F.A.: On Optimization of CSMA/CA based Wireless LANs: Part I – Impact of Exponential Backoff. In: Proc. of ICC (June 2006)

Investigation of Power-Aware IEEE 802. 11 Performance in Multi-hop Ad Hoc Networks

Basel Alawieh[1], Chadi Assi[1], and Hussein Mouftah[2]

[1] Concordia University
Montreal, Quebec, Canada
b_alawi,assi@encs.concordia. ca
[2] University of Ottawa
Ottawa, Ontario, Canada
mouftah@site.uottawa.ca

Abstract. The choice of transmit power determines the transmission range and carrier sensing range. Thus, the transmit power is a key to the tradeoff between the amount of spatial reuse and probability of collisions in wireless multi-hop ad hoc networks. In this paper, we propose a realistic analytical model to study the performance of the IEEE 802.11 access method in multi-hop wireless networks by incorporating the effect of transmit power and physical carrier sense. Using the model, we investigate the effect of these attributes on the throughput obtained by individual nodes. Our results show that performing power control and carrier sense threshold tuning using the two way handshake offer higher advantages and achieve better throughput than that achieved using the four-way handshake.

Keywords: power control, IEEE 802.11, modeling, simulation.

1 Introduction

The rapid evolution of the mobile internet technology has provided incentives for building efficient multi-hop wireless networks. A wireless ad hoc network precludes the use of a wired infrastructure and allows hosts to communicate either directly or indirectly over radio channels. These networks are applicable to environments in which a prior deployment of network infrastructure is not possible. Due to the scarce wireless channel resources, an effective medium access control (MAC) protocol which regulates nodes access to the shared channel(s) is required. Thus, recently the development of efficient MAC protocols for multi-hop ad-hoc networks has been the focus of extensive research. Currently, the IEEE 802. 11 distributed coordination function (DCF) standard [1] has gained global acceptance and popularity both in wireless LANs and wireless multihop ad hoc environment. In IEEE 802. 11 with 4-way handshake, all packets (RTS/CTS/DATA/ACK) are sent with maximum power [5]. This kind of handshake communication has shown to increase the throughput by trying to avoid the hidden terminal at best at the cost of enhancing the exposed terminal problem and therefore, decreasing the spatial reuse (i. e. , suppressing

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 409–420, 2007.

allowable simultaneous transmissions) [5]. Four major directions have been proposed inorder to enhance the spatial reuse for IEEE 802.11. Namely, i) transmit power control [5] ii)data rate adaptation [?] iii) physical carrier sensing threshold tuning [?] iv) binary exponential backoff mechanisms [3].

Yang et al. [8] proposed an two-way handshake model to investigate the impact of transmit power and carrier sense threshold on network throughput in a two-way handshake mechanism; Through their model, the authors argue that an optimum throughput can be achieved for a specific carrier sensing threshold. Moreover, they concluded that higher system throughput can be achieved with the use of smaller transmit power (subject to network connectivity) and carrier sense threshold. Gobriel [4] et al. propose a power-aware four-way handshake model to analyze the maximum throughput and the consumed energy under maximum interference. Nevertheless, in their model the interference range is either not considered or assumed to be fixed and equal to the transmission range. What would be interest, is to investigate the impact of transmit power on throughput for four way handshaking taking into account the effect of carrier sense mechanism and the interference range and compare it to the two-way handshake.

We start by first proposing a realistic analytical model for a wireless multihop network to study the impact of controlling the transmit power on the achievable throughput with four-way (RTS/CTS/DATA/ACK) IEEE 802.11 DCF as the MAC protocol. Furthermore, and through using the derived model, we investigate the interplay between tuning the carrier sensing threshold and transmit power on the achievable throughput. The remainder of the paper is organized as follows. In Section II, we present background on the communication model adopted. Then, we introduce the the model in section III. Analytical and simulation results are presented and argued in section IV. Finally, we conclude the paper.

2 Communication and Interference Models Background

The carrier sensing range and transmission range in wireless ad hoc networks are related according to the carrier sense threshold and reception power threshold, and path loss exponent of the signal:

$$r_c = r_t \cdot C \tag{1}$$

Here, $C = (\frac{\kappa}{\eta})^{\frac{1}{\alpha}}$. κ is the reception sensitivity and η is the carrier sensing threshold.

In our work, we assume that RTS/CTS messages are sent with maximum power to cover a range a range r_t ($r_t = a_{RTS} = a_{CTS}$), while DATA/ACK are sent with power value that is less than the the maximum power to cover distance between the sender and the receiver range ($r_t = r = a_{data} = a_{ack}$). Hence, we have two carrier sensing ranges according to equation (1). These are ($r_c = r_{cRTS} = r_{cCTS}$) for RTS/CTS packets and ($r_c = r_{cdata} = r_{cack}$). Now assuming a fixed bit error rate, we express the power relation between the control messages (RTS/CTS) and the DATA/ACK packets values using the path loss law [4] as:

$$P_{data|ack} = P_{RTS|CTS} \cdot (\frac{a_{data}}{a_{RTS}})^{\alpha} \tag{2}$$

We now define the interference range of a node receiving a packet. Here, the interference range is defined according to [7] as the maximum distance at which the receiver will be interfered with by another source (i. e. signal to interference plus noise ratio at the receiver drops below the threshold value ζ(SINR threshold). Assume an ongoing communication between nodes A and B that are r distant apart, and assume also an interfering node C which is d meters away from node B initiates a communication. Let P_r denote the receiving power of a signal from node A transmitting at power P_{tA} and P_i denote the receiving power of interference signal at the receiver B from node C transmitting at P_{tC}. Neglecting the thermal noise power, and assuming a homogenous network, (all nodes uses the same SINR requirements, and physical medium) and through applying the path loss law, we have:

$$SINR = \frac{P_r}{P_i} = \frac{P_{tA} \cdot r^{-\alpha}}{P_{tC} \cdot \ d^{-\alpha}} \geq \zeta \tag{3}$$

We define the interference set of a receiver B (denoted as IN_B) as the set of nodes whose transmission, if overlapping with the transmission of transmitter s, will cause collision at the receiver. Specifically,

$$IN_B = \{F \mid \frac{P_{tA} \cdot r^{-\alpha}}{P_{tC} \cdot \ d^{-\alpha}} < \zeta\} \tag{4}$$

In order to determine the interference range at node B, we assume the worst case interference scenario; that is the interfering node C is sending either an RTS or CTS packet (i.e., maximum power). With the condition of the interference set from equation (4), and assuming node A sends an RTS packet at maximum power and using equation (3), we define the interference range r_i as the maximum value of d such that equation (6) holds as

$$r_i = \zeta^{\frac{1}{\alpha}} \cdot a_{data} \tag{5}$$

Assuming a successful RTS/CTS handshake, node A sends DATA packet at minimum power and by substituting equation (2) into equation (4), the interference range can be re-derived as:

$$r_i = \zeta^{\frac{1}{\alpha}} \cdot a_{RTS} \tag{6}$$

If we adopt a channel rate of 2 Mbps with ζ equals to 10 dB, we get $r_i = 1.78 \cdot a_{data}$ for RTS or CTS receiver and $r_i = 1.78 \cdot a_{RTS}$ for DATA or ACK receiver.

3 Power-Aware IEEE 802.11 Model

We assume that the system time is slotted with σ seconds as in [2]. RTS, CTS, DATA and ACK packets are assumed to be with fixed length of L_{RTS}, L_{CTS},

L_{DATA} , L_{ACK} bits. The Channel bit rate is assumed to be R bps. Thus the transmission of an RTS/CTS/DATA/ACK packets will last for $T_r = \frac{L_{RTS}}{R \cdot \sigma}$, $T_c = \frac{L_{CTS}}{R \cdot \sigma}$, $T_d = \frac{L_{DATA}}{R \cdot \sigma}$, $T_a = \frac{L_{ACK}}{R \cdot \sigma}$ respectively. Each node in the network initiates a transmission to one of its neighbors at the beginning of a virtual time slot for a duration of T_{avg}. Here, T_{avg} is the interval between the occurrences of two specific events as defined by [2]. We also assume that the network operates under the saturation condition. Now we proceed to model the channel activities from the perspective of an individual sender in order to be able to derive the single node throughput S. Before finding S, a parameter of interest to derive is the attempt probability τ that a sender transmits in a randomly chosen (virtual) slot. Here, τ will be given by [2]:

$$\tau = \frac{2 \cdot (1 - 2 \cdot P_c) \cdot (1 - p)}{(1 - 2 \cdot P_c) \cdot (W + 1) + P_c \cdot W \cdot (1 - (2 \cdot P_c)^m)} \tag{7}$$

p is the probability that the channel is sensed busy and is given by $p = 1-(1-\tau)^{N_c}$, where $N_c = \rho \cdot \pi r_c^2$ is the number of nodes in the carrier sensing range of the transmitter and ρ is the active node density. $m = log_2(CW_{max}/CW_{min})$ where CW_{min} , CW_{max} being the minimum contention window and maximum contention window. Here, P_c is the conditional collision probability and is calculated as follows:

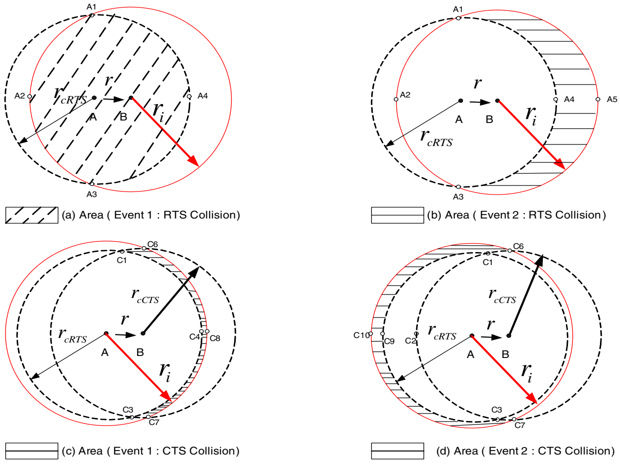
$$P_c = 1 - (1 - p_{rts}) \cdot (1 - p_{cts}) \cdot (1 - p_{data}) \cdot (1 - p_{ack}) \tag{8}$$

$p_{rts}, p_{cts}, p_{data}, p_{ack}$ are the probabilities of RTS, CTS, DATA, and ACK collision respectively.

3.1 RTS/CTS Handshake Collision Events

We now consider the RTS/CTS handshake is taking place between nodes A and node B separated from each other by distance r as shown in Figure 1. Four cases needed to be investigated and these are $r_c \geq r_i + r$, $r_i \geq r_c + r$, $r_c \geq r_i$, $r_c < r_i$; here, $r_c = r_{cRTS} = r_{cCTS}$ and r_i as in equation (5). In this work and due to lack of space, we analyze the case $r_c < r_i$ since in this case we would be able to illustrate more delicately the reasons behind the RTS/CTS collisions (as will be shown shortly). Other cases, $r_c > r_i + r$, $r_c > r_i$ and $r_i > r_c + r$ will follow the same line of analysis.

- *RTS Collision Events* are the various events behind the failure of an RTS packet transmission that yield to an unsuccessful transmission. Node A is initiating an RTS transmission towards node B (Figure 1): In order to determine the probability of RTS packet collision, we need to consider the following two events:
  1. Event 1-RTS: There are one or more transmissions initiated from nodes N_{ci} located in the intersection of the interference range area of the receiver (node B)and the carrier sensing range of the transmitter node A) the area A1A2A3A4 as shown in Figure 1(a)) at time t_r, the time an RTS packet initiated and it is the first time slot in T_r. These nodes will sense



▨ (a) Area (Event 1 : RTS Collision)

▤ (b) Area (Event 2 : RTS Collision)

▤ (c) Area (Event 1 : CTS Collision)

▤ (d) Area (Event 2 : CTS Collision)

Fig. 1. RTS Collision Events

the channel busy if they do not transmit at the first time slot. Assuming that the nodes in the carrier sensing area will sense the same channel status as that sensed by transmitter (node A), then the probability P_{rts1} of this event can be approximated by:

$$p_{rts1}(r) = 1 - (1-\tau)^{N_{ci}(r)} \tag{9}$$

if we denote $A(r)$, the area where N_{ci} nodes are located, then $N_{ci}(r) = \rho \cdot A(r)$.

2. Event 2-RTS: A collision occurs if there are one or more transmissions initiated during the vulnerable interval $[t_r - T_r + 1, t_r + T_r - 1]$ from nodes N_h located in the hidden nodes area as shown in Figure 2(b) (A1A4A3A5A1). Those nodes if they transmit during the vulnerable period (period during which data packet is transmitted), a collision will occur. The probability of this event p_{rts2} is

$$p_{rts2}(r) = 1 - (1-\tau)^{N_h(r) \cdot \frac{T_r}{T_{avg}}} \tag{10}$$

where $\frac{T_r}{T_{avg}}$ is the number of virtual slots in the vulnerable period. Since at the beginning of each of these virtual slots a node may attempt for transmission, the term $(1-\tau)^{N_{h(r}}$ has to be raised to a power of $\frac{T_r}{T_{avg}}$.

Therefore, the probability of RTS collision (p_{rts}) is simply:

$$p_{rts}(r) = 1 - (1 - p_{rts1}(r)) \cdot (1 - p_{rts2}(r)) \tag{11}$$

- *CTS Collision Events* are the events that a collision occurs during a CTS packet transmission. Assuming RTS packet has succeeded, node B will reply with a CTS. Nodes within the carrier sensing zone of node A and outside the transmission zone will defer their transmission to EIFS duration. Thus, these nodes will not interfere with the CTS reception at node A since EIFS is a sufficient period for CTS to be transmitted and received at the sender. Two reasons exist for a CTS packet to collide.

1. Event 1-CTS: There are one or more transmissions initiated from nodes N_{cci} as shown in Figure 1(c) located in the area (C1C6C8C7C3C4C1)at t_c (the time a CTS packet initiated and it is the first time slot in T_c). Following the same assumption as event 2 for RTS packet collision, we derive p_{cts1}:
$$p_{cts1}(r) = 1 - (1-\tau)^{N_{cci}(r)} \tag{12}$$
2. Event 2-CTS: A collision occurs if there are one or more transmissions initiated during the interval T_c (vulnerable period) from nodes N_{hct} (as shown in Figure 1(d)) located in the hidden area (C1C6C10C7C3C9C1) any time slot within $[t_c - T_c + 1, t_c + T_c - 1]$. The probability $P_{cts2}(r)$ of this event is:
$$p_{cts2}(r) = 1 - (1-\tau)^{N_{hct}(r)\cdot \frac{T_c}{T_{avg}}} \tag{13}$$

Therefore, $p_{cts}(r)$ is simply :

$$p_{cts}(r) = 1 - (1 - p_{cts1}(r)) \cdot (1 - p_{cts2}(r)) \tag{14}$$

3.2 DATA/ACK Handshake Collision Events

Now node A and node B decided to reduce the transmit power for DATA-ACK so as only to cover the distance between them as discussed earlier. hence, the transmission range ($r_t = a_{data} = a_{ack}$) for DATA-ACK becomes smaller than that of RTS-CTS ($r_t = a_{RTS} = a_{CTS}$). Similarly, the new carrier sensing range r_{cdata} for DATA-ACK also becomes smaller (since it will be a function of a_{data} according to equation (1)) than that of RTS-CTS. Now, depending on the distance between nodes A and B ($r = a_{data}$), and assuming a successful RTS/CTS handshaking has taken place, four scenarios should be distinguished in order to determine the collision probabilities of DATA/ACK: $r_c > r_i$, $r_c \geq r_i + r$, $r_c < r_i$, $r_i \geq r_c + r$. Here, $r_c = r_{cadata} = r_{cack}$ and r_i as derived in equation (6). Similarly, we analyze the case $r_c < r_i$. The other cases follow the same line of analysis.

Now, we distinguish two types of DATA collision events that may occur.

- Event1-DATA: Nodes N_{hd} that mainly lie in the carrier sensing zone of node B when transmitting the CTS message may become active after EIFS and can start transmission during any time slot within $[t_d - T_d + 1 - EIFS, t_d + T_d - 1 - EIFS]$, t_d (the time node A starts transmitting its data packet) and thus may corrupt the packet reception at node B. So, the probability p_{data1} of this event is
$$p_{data1}(r) = 1 - (1-\tau)^{N_{hd}(r)\cdot \frac{T_d - \frac{EIFS}{\sigma}}{T_{avg}}} \tag{15}$$
- Event2-DATA: Hidden nodes N_{hd1} that are mainly located outside the carrier sensing zone of node B when transmitting the CTS message and

within the interference range can start transmission any time slot within $[t_d - T_d + 1, t_d + T_d - 1]$. The probability p_{data2} of this event

$$p_{data2} = 1 - (1 - \tau)^{N_{hd1}(r) \cdot \frac{T_d}{T_{avg}}} \tag{16}$$

$p_{data}(r)$ can be calculated as follows:

$$p_{data}(r) = 1 - (1 - p_{data1}(r)) \cdot (1 - p_{data2}(r)) \tag{17}$$

The calculation of the number of hidden nodes N_{hd} and N_{hd1} are dependent on the transmission power of the data packet. This is what we are going to elaborate shortly. If the DATA packet is transmitted at a power such that $r_i > r_{cdata}$, two different scenarios are to be considered within this case: $r_{cCTS} \geq r_i$, (r_{cCTS} is the carrier sensing range of the transmitted CTS packet and depends on the power at which the CTS is transmitted) and $r_{cCTS} < r_i$.

- $r_{cCTS} \geq r_i$: In this case, the probability of data packet collision will result from Event1-DATA. Within this scenario, we distinguish the following cases: $r_{cdata} \geq a_{CTS} + r$ or $r_{cdata} > a_{CTS}$;
  1. $r_{cdata} \geq a_{CTS} + r$ (Figure 2(a)): In this case,the collision probability for data packet, p_{data1}, will be the result of nodes N_{hd} existing in the shaded area (A1A2A3A4A1).
  2. $r_{cdata} > a_{CTS}$ (Figure 2(b)): p_{data1} for this case will be the result of nodes N_{hd} existing in shaded area (A1A2A3A4A5A6A1).

- $r_{cCTS} < r_i$: Both p_{data1} and p_{data2} exist for this scenario and we distinguish between the following: $r_{cdata} \geq a_{CTS} + r$ or $r_{cdata} > a_{CTS}$ to determine p_{data1} ;

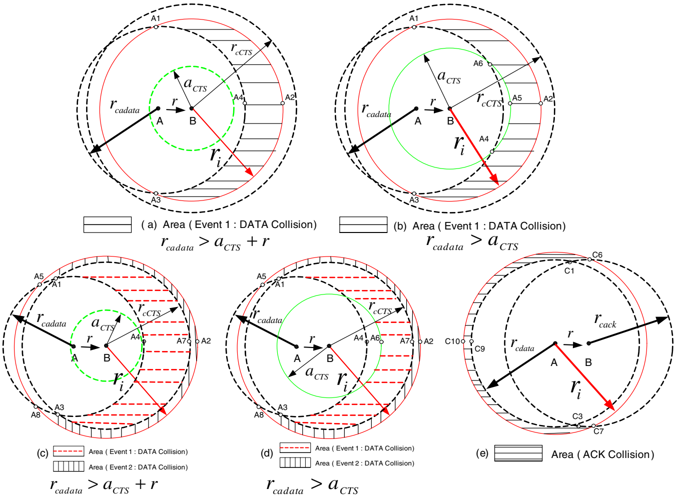
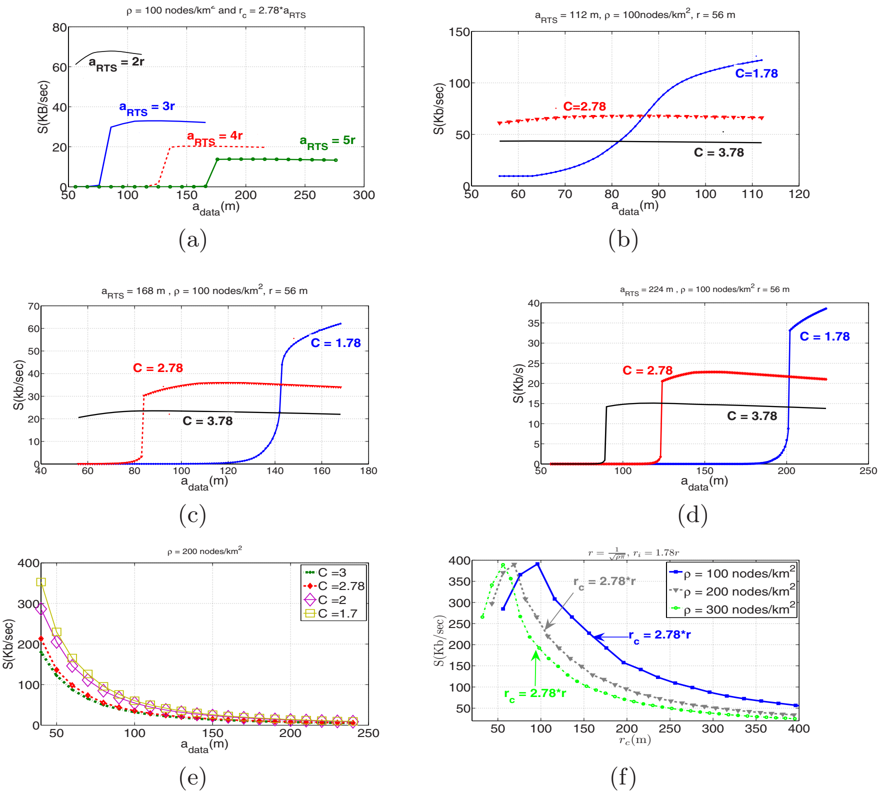


Fig. 2. DATA/ACK Events

1. $r_{cdata} \geq a_{CTS} + r$:(Figure 2(c)) The collision probability for data packet, p_{data1}, will be the result of nodes N_{hd} existing in shaded area (A1A4A3A7A1) .
2. $r_{cdata} > a_{RTS}$:(Figure 2(d)) will be the result of nodes N_{hd} existing in shaded area (A1A7A3A6A1).

p_{data2}, on the other hand, will be the result of nodes N_{hd1} existing in shaded area (A1A5A2A4A3A7A1) Figure 2(c)-(d).

We proceed now to define the ACK collision event and determine the probability of ACK event accordingly. Since nodes in the carrier sensing zone of node A defers for EIFS, the ACK packet can be correctly received by node A without any interference from these nodes (in the carrier sensing zone of node A). This due to the fact that EIFS duration is sufficient time for an ACK packet to be transmitted. Hidden nodes,N_{hack}, that may interrupt the ACK reception exist only in the case when r_i is larger than r_{cdata}. Since we are considering $r_i > r_{cdata}$ scenario, the nodes N_{hack} that might interfere while the ACK packet is received



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 3. Analysis of the effects of transmit power and carrier sensing threshold

at the sender will be those located in area (C1C6C10C7C3C9C1C6) as shown in Figure 2(e). The probability for ACK packet collision , p_{ack}, is:

$$p_{ack}(r) = 1 - (1 - \tau)^{N_{hack}(r) \cdot \frac{T_a}{T_{avg}}} \tag{18}$$

The throughput, (S) for each transmitter is calculated as follows:

$$S = \frac{P_t \cdot L_{DATA}}{T_{avg}} \tag{19}$$

where P_t is the probability that a node makes a transmission attempt and the attempt is successful. Hence P_t is given by $P_t = \tau(1 - P_c)$. We analyze four possible channel activities from the perspective of a node attempting to send a packet in a virtual slot in order to determine T_{avg}.

– idle: This indicates that there is no transmission on the channel (i. e. , the received power level is below the carrier sense threshold η). The probability of the virtual slot to be an idle slot is simply $P_i = (1 - \tau)^{N_c+1}$. That is none of the nodes in the carrier sensing range of the transmitter transmits at this time slot. The duration of this slot is $T_i = \sigma$.
– collision: If the transmitter does not receive a CTS (ACK) within an interval of SIFS after the RTS (DATA) frame is transmitted, it determines that a collision has occurred. The probability that a virtual slot is a collision slot is τp_{col} with a duration T_{col}.
– Success: This activity shows that node A has received an ACK frame within an interval of SIFS after the DATA frame is transmitted, for which it determines that the transmission is successful. The duration of this activity is $T_{transmit}$; the probability of the virtual slot is success slot is P_t.
– busy: A node in its backoff stage attempting to transmit a packet, will sense the channel busy. The probability of the virtual slot to be a busy slot with duration T_{busy} is simply $(1 - \tau)p$. Here, $T_{busy} = (1 - P_c) \cdot T_{transmit}$ is approximated on the assumption that a node will always defer if the channel is sensed busy and assuming an average collision probability for all nodes.

Based on the above analysis , T_{avg} can be calculated as follows:

$$T_{avg} = P_i T_i + P_t T_{transmit} + \tau p_{col} T_{col} + (1 - \tau) p T_{busy} \tag{20}$$

where $p_{col}T_{col} = (1 - (1 - p_{rts})(1 - p_{cts}))T_{rr} + (1 - (1 - p_{data})(1 - p_{ack}))T_{dd}$. Here, $T_{transmit}$, T_{rr}, T_{dd} are respectively the successful time need for successful data packet delivery, RTS timeout duration, and DATA timeout duration and these value can be found from IEEE 802.11 standard [1].

4 Performance Study

The model parameters are taken from the IEEE 802. 11b specifications [1]. The parameters for analysis and simulations are SIFS: 10 μs, DIFS = 50 μs, EIFS = 364 μs, σ = 20 μs, Basic rate = 2 Mb/sec, PCLP Length = 192 bits @ 1 Mbps,ζ

= 10dB, Packet size = 1000 bytes, $[CW_{min}, CW_{max}] = [31, 1023]$. Here, the term $r = \frac{1}{\sqrt{\rho\pi}} = 56$ for $(\rho = 100 nodes/km^2)$ which is the average distance for receiver to exist within the transmission range of the sender will be used in the analysis, in which data transmission range a_{data} and control transmission range are given, if otherwise specified. Here, r is the lower bound of a_{data}. Moreover, we analyze and refer to a_{RTS} in this text for both RTS/CTS transmission ranges since they are equal; this applies also to DATA/ACK transmission ranges a_{data}. We start by first showing the throughput performance as we vary the transmission power for DATA packets (i.e., $a_{data} \leq a_{RTS}$) and for different transmission ranges a_{RTS} for RTS/CTS packets . Figure 3(a) clearly shows that the highest throughput is obtained for shorter ranges of RTS (i.e., lower power for RTS packets), while the throughput decreases for longer ranges for RTS packets. This is due to the fact that when a_{RTS} is smaller, the interference range for DATA packets (equation 6) (interference area around the receiver when the transmitter is sending its DATA packet) decreases, thereby reducing the impact of hidden terminals on the reception of DATA. However as a_{RTS} increases, the interference range for DATA (equation 6) increases as well, leading to more DATA collisions and hence reduced throughput. In addition, increasing the transmission range a_{RTS} will silence more nodes(that falls in the carrier sensing range of the transmitter and may not be in the interference range of the receiver) thus it has the effect of reducing the spatial reuse, leading to lower throughput. Note that at a fixed a_{RTS} and a lower a_{data}, the network exhibits low performance; this is because the interference range for DATA reception is large while the carrier sensing range(r_{cdata}) when transmitting the DATA packet is small. Hence, the area of interference that is not covered by the carrier sensing range (r_{cdata}) is large which leads to higher DATA packet collisions (i.e., nodes that can corrupt the reception of the DATA packet and cannot be silenced by the transmitter). On the other hand, increasing a_{data} will increase the carrier sensing range (r_{cdata}) and reduce the area of the interference range (equation 6) that is not covered by the carrier sensing range (r_{cdata}). Accordingly, the DATA packet collision incrementally reduces, leading to a better system throughput. Moreover, with further increase in a_{data}, the throughput will slightly decrease. This is due to the fact as a_{data} approaches a_{RTS}, the carrier sensing range (r_{cdata}) covers entirely the interference around the receiver and becomes large enough to suppress other nodes from communicating (i.e., impair the spatial reuse). We elaborate more on this through an example where we set $a_{RTS} = 2r$, as shown in Figure 3(a).. In this case the interference range around the data receiver is $r_i = 3.56r$ where as the carrier sense range $r_{cdata} = 2.78a_{data}$, so through the increase of a_{data} from r towards $2r$, there comes a point when r_{cdata} becomes equal to $r_i + r$, in such a case the probability of collision is zero, so the increase further beyond this point will definitely decrease the throughput since more nodes will be unnecessary silenced. We can see the selection of the carrier sensing threshold and the threshold of the RTS/CTS packet determine these operating condition, for example with $C = 1.78$, we will not get the operating condition as that of $C = 2.78$. We more validate on this in the next experiment.

Next we study the network performance as we vary the carrier sensing threshold (i.e. C in equation 1). Figure 3(b) shows the node's throughput for a fixed a_{RTS} and while varying the transmit power for DATA packets. Figure 3(b) shows when the carrier sensing threshold ($C = 1.78$) is small and at smaller transmit power for DATA packet, the system yields a very poor performance. This is because the data interference range (r_i(equation (6)) is large enough while the carrier sensing range (r_{cdata}) is very small which means the hidden terminals adversely affect the network. As a_{data} increases, the carrier sensing starts to cover the interference range of the receiver until ultimately covers the whole interference area and silences all the hidden nodes (at the expense of amplifying the exposed terminals). Clearly, the effect of the exposed terminals is not as strong as that of the hidden nodes (it is verified through numerical results that the collision probability of DATA packets is reduced significantly). At larger carrier sensing thresholds (e.g., $C = 2.78$), we observe that at lower a_{data}, the node's throughput is improved (as opposed to the case where $C = 1.78$); this is due to the fact that the carrier sensing range is increased thereby covering (partially or completely) the interference area of the receiver. Further increasing C (e.g., $C = 3.78$) will result in a very large carrier sensing area that will completely cover the interference area of the receiver and severely affect the spatial reuse in the network. Therefore, the reduction in the node's throughput as C increases. Similar observations have been seen for larger values for a_{RTS} (Figures 3(c), 3(d)).

Now we study the performance of the two-way handshaking mechanism. The events for RTS and CTS are used for the DATA and ACK. We fix the carrier sensing threshold C , and vary a_{data} accordingly. Figure 3(e) shows that as a_{data} increases the throughput decreases. This is due to the fact that at larger a_{data}, the interference range (r_i -equation 5) increases and hence the interference area increases. Consequently, the data collision probability increases which in turn leads to the decrease in system throughput. Furthermore, the increase of a_{data} silence other nodes (i.e., impair the spatial reuse). We can also infer from Figure 3(e) that for fixed a_{data}, an optimum carrier sensing threshold is attained and that it is not at $C = 2.78$. Here, when $C = 2.78$ that is ($r_{cdata} = r_i + r$) , the effect of hidden nodes is completely eliminated. Figure 3(f) proves this claim [8]. Figure 3(f) shows for fixed r (thus fixed r_i), as we vary the carrier sensing threshold such that r_{cdata} to become much larger than $r_i + r$, the throughput decreases since more nodes will be silenced. Moreover as r_{cdata} becomes smaller than $r_i + r$, the throughput increases then decreases. This is due to the fact that, when r_{cdata} becomes sufficiently small, the area resulting from the interference range minus intersection between the interference range and carrier sensing range (the area in the event-2 RTS section 3.1) becomes large even though the area of intersection between the interference range and carrier sensing range (the area in the event-1 RTS section 3.1) becomes small. As a result, the term $N_{hd}\frac{T_d}{T_{avg}}$ becomes dominant because of the factor $\frac{T_d}{T_{avg}}$. Thus, the collision probability increases, and the throughput decreases. Now we compare the performance of two-way handshaking with four-way handshaking in a saturated network and under worst-case scenario. We can see that the maximum throughput (C= 2.78) attained for two-way handshaking is around 220 Kb/sec while that for four-way handshaking is almost 70 Kb/sec for the case $a_{RTS} = 2r$ (even when sending $a_{RTS} = a_{data} = r$, throughput attained is almost 160 Kb/sec) (we omit the other scenario results due to lack of space). Thus performing power control using two-way handshaking is much beneficial than four-way handshake for worst-case scenario. Finally, we conducted extensive discrete event simulations via Qualnet [6] to study the validity of the proposed analytical model. Analytical results match reasonably well with simulation results with a maximum error of 10 %.

5 Conclusions

The goal of this paper was to develop a methodology, and some insights, on how ad hoc network capacity is affected by the physical attributes (transmit power and carrier sensing threshold). We verified that sending RTS/CTS at maximum power and DATA/ACK at minimum power may not be an efficient scheme; we showed that for large packet sizes when carrier sensing threshold is very low, varying the transmission power of data packet does not add any benefits, whereas when increasing the carrier sensing threshold, it is more advantageous to perform power control Moreover, we conclude that performing power control using the two-way handshaking is much better than using four-way handshaking.

References

1. IEEE 802.11 wireless LAN media access control (MAC) and physical layer (PHY) specifications (1999)
2. Bianchi, G.: Performance Analysis of the IEEE 802.11 distributed coordination function. In: IEEE JSAC (March 2000)
3. Durvy, M., Thiran, P.: A Backoff Mechanism to Achieve Full Organization. In: SpaSWiN (2007)
4. Gobriel, S., Melhem, R., Mossé, D.: A Unified Interference/Collision Analysis for Power-Aware adhoc networks. In: INFOCOM (2004)
5. Muqattash, A., Krunz, M.: A Single-Channel Solution for Transmit Power Control in Wireless Ad Hoc Networks. In: MobiHoc (May 2004)
6. Qualnet Simulator, www.scalablenetworks.com
7. Xu, K., Gerla, M., Bae, S.: Effectiveness of RTS/CTS hanshake in IEEE 802.11 based ad hoc networks. Journal of Ad hoc networks, 107–123 (2003)
8. Yang, Y., Hou, J.C., Kung, L.-C.: Modeling the Effect of Transmit Power and Physical Carrier Sense in Multi-hop Wireless Networks. In: INFOCOM (2007)

Access Scheduling on the Control Channels in TDMA Wireless Mesh Networks

Hongju Cheng[1], Xiaohua Jia[2], and Hai Liu[2]

[1] College of Mathematics and Computer Science, Fuzhou University
Fuzhou, P.R. China
cscheng@fzu.edu.cn
[2] Department of Computer Science, City University of Hong Kong,
83 Tat Chee Ave, Kowloon, Hong Kong
{jia, liuhai}@cs.cityu.edu.hk

Abstract. The access scheduling on the control channels in TDMA wireless mesh networks is studied in this paper. The problem is to assign time-slots for each node in the network to access the control channels so that it is guaranteed that each node can transmit a control packet to any one-hop neighbor in the scheduling cycle. The objective is to minimize the total number of different time-slots in the cycle. This paper has taken the large interference range problem into consideration and proposed two algorithms for the scheduling problem, namely, One Neighbor Per Cycle (ONPC) algorithm and All Neighbors Per Cycle (ANPC) algorithm. The number of time-slots by ANPC algorithm is upper-bounded by min(n, $4K - 2$) in some special cases, where n is the node number and K is the maximum node degree. Both centralized and fully distributed versions are given. Simulation results also show that the performance of ONPC is rather better than ANPC.

Keywords: Access Scheduling, Control Channel, TDMA (Time-Division-Multiple Access), Wireless Mesh Networks, Algorithms.

1 Introduction

The wireless mesh network [1] is designed to support wireless broadband access and various scenarios. The QoS (Quality-of-Service) support is a key issue in these applications. When a node wants to exchange the control packet with its neighbors, it is required that the exchange process will finish within a guaranteed delay bound. The contention-based access scheduling, such as the IEEE 802.11 DCF, cannot provide such guarantee because of its random contention and back-track mechanism.

It leads to a determined access scheduling in which each node is assigned with one / several chance(s) (time-slots in a TDMA system) to access the control channel, and the transmission of control packets on the assigned time-slots is guaranteed to be received collision-free by the neighbor(s). The access delay on the control channels is upper-bounded by the length of the scheduling cycle. Note that a single time-slot can be reused by two nodes if they do not interfere with each other.

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 421–432, 2007.

When a dedicated channel is shared by all nodes to transmit control packets, the cycle length can be too long because all nodes contend on it [2, 3]. Vaidya et al. had proposed a multi-channel model [4] where the mesh node can switch its radio among available channels. If one node needs to send control packets to another, it switches its radio to the fixed channel of the target and sends on it. Note a node always listens on its fixed channel. Switching a radio from one channel to another incurs some delay in range of a few hundred microseconds to a few milliseconds [16, 17].

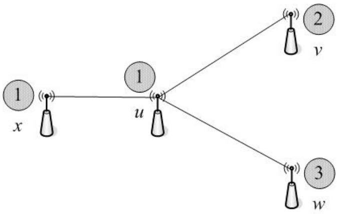


Fig. 1. The control packet exchange mechanism in the network

Figure 1 shows an example of this hybrid multi-channel model. The number in the gray circle is the fixed channel of the node. In this example the fixed channels for *u*, *v*, *w*, and *x* are channel 1, 2, 3, and 1 accordingly. *u* and *v*, *u* and *w*, and *u* and *x* can communicate with each other directly. *u* can switch to the fixed channel of *v*, i.e., channel 2, and send the control packet to *v*. Similarly, *u* also can switch to channel 1 and send the control packet to *x*.

This hybrid multi-channel model can reduce congestion on the control channels. In this paper we have proposed two access scheduling algorithms. Our algorithms have the following advantages: 1) A determined scheduling which improves the system performance by eliminating all conflicts and retransmissions in the network; 2) Guaranteed access delay with upper-bounded access delay; 3) Easy to be realized in a fully distributed way; 4) High utilization with load balance on the channels.

The rest of this paper is organized as follows. In section 2 we discuss the related works. In section 3 are the system model and problem formulation. In section 4 we propose two centralized algorithms. In section 5 we introduce fully distributed versions of these algorithms. Simulation results are presented in section 6. And in section 7 are concluding remarks.

2 Related Works

So far as we know, this is the first paper on the access scheduling problem in multi-channel mesh networks. The known related works are all done in the single channel wireless networks and on the assumption that the interference range is equal to the transmission range which is proved not true [6].

Chlamtac and Pinter [2] have proposed a greedy algorithm for the access scheduling problem in single channel networks. The basic idea is that each node selects a time-slot for broadcasting with 2-hop local information. And the number of time-slots is upper-bounded by min(n, K^2+1), where n is the node number and K is the maximum node degree. The main problem is that the interference range is assumed equal to the transmission range. However, recent results [6] have shown that the interference range is generally about 2~3 times the transmission range when the distance of the transmitter and receiver is close to the transmission range.

Some algorithms make access scheduling on the channel with in-band signaling, such as phase negotiation and RTS/CTS handshakes [7-9]. The in-band signaling is very simple and easy to realize, but it cannot fully eliminate the potential interference because the interference range can be larger than the transmission range, and thus collision may occur at some time-slots. Another problem is that these methods spend some band resource on contentions. Further, the deadlock may happen in the phase negotiation method because nodes know nothing about the local topology.

A number of topology-independent scheduling algorithms have been proposed [10-12]. The basic idea of the topology-independent scheduling is for a node to transmit in several time-slots in one scheduling cycle so that there is a unique time-slot on which a node can communicate with any one-hop neighbor. The benefit of the topology-independent scheduling is that the scheduling result is independent to the topology of the network and thus very suitable for dynamic scenarios where the topology changes frequently. The limitations are that the length of the scheduling cycle is generally very large and the result always depends on the network size.

The broadcast scheduling problem is studied in [5, 13-15]. Most works are centralized algorithms [13-15]. Ramanathan and Lloyd [5] have proposed several on-line algorithms. The on-line algorithms need a centralized node to deal with these events and thus they are in fact centralized. There is no fully distributed algorithm for the broadcast scheduling problem till now.

3 System Model and Problem Formulation

Assume there are totally Q channels in the network numbered from 1 to Q. For each node u, there is a pre-assigned control channel called as the *fixed channel* which is notated as *fixed*(u). Different nodes may have different fixed channels. The network topology can be represented by a graph $G = (V, E)$ where V is the set of nodes and E is the set of edges. There is an edge (u, v) between node u and v if they can communicate with each other, or their Euclidean distance is no large than the transmission range. We assume all nodes have the same transmission range.

The number of edges on the shortest path $u = v_0, v_1, \ldots, v_h = v$ is defined as the distance between node u and v. The set of nodes whose distance to node u is no more than h is defined as the h-hop neighbors of node u, i.e., N_u^h. Especially N_u^1 denotes the set of one-hop neighbors of node u in the network.

Recent results [6] have shown that the interference range is generally larger than the transmission range, that is, about 2~3 times when the distance of the transmitter and receiver is close to the transmission range. Here we propose an H-hop interference model to describe this large interference range problem which is defined as below. Supposing node u transmits over channel k to node v, this transmission is successfully received by node v if no other node $w \neq u$ within H-hop from v is transmitting over channel k simultaneously. The H-hop interference model has described the situation where a zone (within H hops from the receiver) is specified to prevent a nearby node from transmitting on the same channel at the same time. The parameter H is a constant depending on the system Signal-To-Interference ratio (SIR) required for a correct reception.

The access scheduling problem is to assign time-slots to each node in the network for the transmission of control packets. Assume one node, say u, is scheduled to send control packets to another node, say v. The following conditions must be satisfied.

$C1$. u should transmit on v's fixed channel.
$C2$. Transmission and reception between u and v must occur in the same time-slot
$C3$. One time-slot can not be used for transmission and reception simultaneously.
$C4$. u can not transmit or receive on another channel at the same time-slot
$C5$. Nodes in N_v^H except u can not transmit on channel k at the same time-slot.

The access scheduling problem on the control channels can be formulated as: given a graph $G = (V, E)$ and *fixed*(u) for each $u \in V$, assign time-slots to each node u in V so that the node can transmit the control packet to any one-hop neighbor in the scheduling cycle. The objective is to minimize the cycle length.

Consider the special case where all nodes in the network have the same fixed channel, a single time-slot cannot be assigned to two nodes if they are within $(H + 1)$-hop. If we view the time-slot as the color, the scheduling problem can be induced to a graph coloring problem as following: construct graph $G' = (V', E')$, in which $V' = V$ and $(u, v) \in E'$ if and only if the distance of u and v on graph G is no more than $(H + 1)$. The graph coloring problem is proved to be NP-complete even in some special graphs. It means that the access scheduling problem on the control channels is also NP-complete. Heuristic algorithms are necessary to solve it in polynomial time.

4 Centralized Algorithms

4.1 The One Neighbor Per Cycle (ONPC) Algorithm

The One Neighbor Per Cycle (ONPC) algorithm is to assign one slot for each node. It is to guarantee that each node reserves the right to transmit a control packet in every cycle. The scheduling also means that all one-hop neighbors shall listen at the same time-slot on their own fixed channel. The strategy is similar to the problem of collision-free scheduling in a single-channel radio network [2, 5, 10, 13-15].

In Figure 1, assume that 2 is the assigned time-slot for u. At time-slot 2, u can switch to channel 1 and send the control packet to x in one scheduling cycle; u also can switch to channel 2 (or 3) and send to v (or w) in another cycle. But u can not speak to both v and x in one cycle because v and x have different fixed channels. Note that v, w and x have to listen at time-slot 2 simultaneously in this case.

The basic idea of the ONPC algorithm is as following. Arrange the nodes by decreasing order of node degree into sequence $\{1, 2, \ldots, n\}$, and then assign time-slots to nodes one by one. The assigned time-slot is the least time-slot unused by its $(H + 1)$-hop neighbors. The process continues until all nodes are assigned. The pseudo code for the algorithm is listed in Figure 2.

Let n be the number of nodes in the network and K be the maximum node degree. The following theorems provide an upper bound for the length of the scheduling cycle by ONPC algorithm.

```
Algorithm Centralized ONPC Time-slot Assignment
Input: topology $G = (V, E)$ and $fixed(u)$, $u \in V$.
Output: the assigned time-slots for each node $u$, $u \in V$.
1. Arrange the nodes by decreasing order of the node degree into sequence
   $\{1, 2, \ldots, n\}$.
2. Select the first unassigned node in the sequence.
3. Select the least number of time-slot that can be used to transmit to all
   one-hop neighbors that satisfies conditions $C1 \sim C5$.
4. If all nodes are assigned, stop. Otherwise, return 2.
```

Fig. 2. The centralized ONPC algorithm

Theorem 1. For arbitrary graphs, the ONPC algorithm guarantees an upper bound of $\min(n, K + 1 + K(K-1)\frac{(K-1)^H - 1}{K-2})$ on the cycle length.

Proof. Suppose the ONPC is to assign one time-slot for u. To satisfy $C1 \sim C5$, the assignment results that might interfere with the assignment of u are those of nodes within $(H + 1)$-hop from u. The number of nodes that are one-hop away from u is no more than K, and the number of nodes that are 2-hop away is no more than $K(K - 1)$, …, the number of nodes that are $(H + 1)$-hop away is no more than $K(K - 1)^H$. The total number of potential interference nodes is:

$$K + K(K-1) + \ldots + K(K-1)^H = K + (K-1)\frac{(K-1)^H - 1}{K-2}.$$

It means that the number of conflicting time-slots is no more than $\min(n - 1, K + K(K-1)\frac{(K-1)^H - 1}{K-2})$. Note the algorithm always assign the least unsigned time-slot, the number of time-slot assigned to u is no more than $\min(n, K + 1 + K(K-1)\frac{(K-1)^H - 1}{K-2})$. □

Lemma 1. For arbitrary graphs, ONPC algorithm guarantees an upper bound of $\min(n, K^2 + 1)$ for the cycle length in case that $H = 1$.

Proof. The conclusion can be easily induced with Theorem 1 with $H = 1$. □

Theorem 2. The time complexity of the ONPC algorithm is $O(n^2)$.

Proof. The time complexity of the ordering is $O(n\log n)$. In the following assignment process, the number of node selection and time-slot assignment is n. The number of comparison is at most n when selecting one time-slot in each assignment. So the full time complexity of the ONPC algorithm is $O(n\log n) + O(nn) = O(n^2)$. □

4.2 The All Neighbors Per Cycle (ANPC) Algorithm

The ONPC algorithm assigns one time-slot to one node. Differently the All Neighbors Per Cycle (ANPC) algorithm assigns several time-slots to one node. The assignment results not only indicate the time-slot location in the cycle, but also the channel that the node should switch to, and accordingly the receivers (the one-hop neighbors with the identical fixed channel).

Let *NFixed*(*u*) be the set of the fixed channels of one-hop neighbors of *u*. The basic idea of ANPC is as following. Arrange the nodes by decreasing order of node degree into sequence {1, 2, …, *n*}. Select the first node *u* in the sequence. For each channel *k* in *NFixed*(*u*), assign one time-slot and ensure that the transmission of control packets at it on channel *k* can be correctly received by all its one-hop neighbors whose fixed channel is *k*. The process continues until all channeles in *NFixed*(*u*) are assigned. Pseudo code for the algorithm is listed in Figure 3.

In Figure 1 *u* has three neighbors as *v*, *w* and *x* with different fixed channels. Assuming the scheduling results for *u* are time-slot 1, 2 and 3, and relevant channel is 1, 2 and 3. Then in one cycle *u* can switch to channel 1 at time-slot 1 and send control packet to *x*. In the same cycle *u* can also switch to channel 2 at time-slot 2 and send control packet to *v*; and channel 3 at time-slot 3 to *w*. In this case *x* only needs to listen at time-slot 1; while *v* listens at time-slot 2 and *v* at time-slot 3.

```
Algorithm Centralized ANPC Time-slot Assignment
Input: topology $G = (V, E)$ and $fixed(u)$, $u \in V$.
Output: the assigned time-slots for each node $u$, $u \in V$.
1. Arrange the nodes by decreasing order of node degree into sequence {1,
   2, ..., n} and mark all nodes as UNASSIGNED.
2. Select the first unassigned node u and calculate its NFixed(u).
3. For each channel number k in NFixed(u), select the least number of
   time-slot that can be used by u to transmit to all neighbors with fixed
   channel as k, and satisfy conditions C1 ~ C5.
4. Change u's state as ASSIGNED.
5. If all nodes are assigned, stop. Otherwise, return 2.
```

Fig. 3. The centralized ANPC algorithm

Intuitively the ANPC needs more time-slots than the ONPC. However, both the theoretic and simulation analysis show that it is not true. The main reason is that if one time-slot is assigned to a node in the ONPC algorithm, and all its one-hop neighbors have to listen at this time-slot. With condition *C*3 and *C*4, this time-slot cannot be reused by all these neighbors. However, the ANPC algorithm has explicitly indicated the receivers at each time-slot, and thus other neighbors can reuse this time-slot and accordingly the total number of time-slots is reduced.

Theorem 3. For arbitrary graphs, ANPC algorithm guarantees an upper bound of $\min(n,\ 2K + 1 + M[(K-1)\frac{(K-1)^H - 1}{K-2} + K - 2])$ for the cycle length, where M is the maximum number of one-hop neighbors with same fixed channels, $1 \le M \le K$.

Proof. Assume that we are to assign one time-slot on channel k for node u. The receivers are the one-hop neighbors whose fixed channel is k. Let S be the set of such nodes and m be the size of S, $0 < m \le M$. Now calculate the number of time-slot that cannot assigned to node u.

To satisfy $C1$ and $C4$, time-slots that have been assigned for transmissions from u to its neighbors can not be reused. The number of such time-slots is at most $(K - m)$. Time-slots that have been assigned for reception of group S from other neighbors can not be reused for this transmission. The number of the time-slots is at most $m(K - 1)$. To satisfy $C2$ and $C3$, time-slots that have been assigned to group S for transmission cannot be reused. The number of such timeslots is at most mK. At same time, time-slots that have been assigned to u for reception cannot be reused. Note that the reception of node u is also the transmission of its one-hop neighbor and the number is partially included above. The excluded number is $(K - m)$ (transmission from the neighbors not in S). To satisfy $C5$, time-slots that have been assigned to group v's H-hop neighbors except u for transmissions on the fixed channel of v can not be reused. The number of such timeslots is at most

$$m(K-1)+m(K-1)^2 \ldots + m(K-1)^H = m(K-1)\frac{(K-1)^H-1}{K-2}.$$

Note that timeslots that have been assigned for reception by group S are also included in above. So the total number of timeslots that are *NOT* available is:

$$K - m + mK + K - m + m(K-1)\frac{(K-1)^H-1}{K-2} = 2K + m[(K-1)\frac{(K-1)^H-1}{K-2} + K - 2].$$

Therefore, timeslot as $\min(n,\ 2K + 1 + m[(K-1)\frac{(K-1)^H-1}{K-2} + K - 2])$ can be assigned for the transmission from u on channel k to group S.

When $m = M$, the length of the cycle is maximized and the theorem stands. □

Lemma 2. For arbitrary graphs, ANPC algorithm guarantees an upper bound of $\min(n, 4K - 2)$ for the cycle length if $H = 1$ and every two neighbors have different fixed channels for any node.

Proof. If every two neighbors have different fixed channels for any node, it means that $M = 1$. The lemma can be induced with Theorem 3 with $H = 1$ and $M = 1$. □

Theorem 4. The time complexity of the ANPC algorithm is $O(Kn^2)$.

Proof. The time complexity of the node ordering is $O(n\log n)$. In the following assignment process, the number of node selection is n and the assignment is at most Kn. The number of comparison is at most n while selecting one time-slot for one assignment. So the full time complexity of the ANPC algorithm is $O(n\log n) + O(Knn) = O(Kn^2)$. □

5 Distributed Algorithms

With the H-hop interference model, the time-slot assignment for one node will influence the assignment for nodes at most $(H + 1)$-hop away. It means that two nodes can be scheduled simultaneously if their distance is more than $(H + 1)$.

The basic idea of the distributed algorithms is as following: each node first collects node degree information of these nodes within (H + 1)-hop away. If a node finds that its node degree is maximal compared with the (H + 1)-hop neighbors which are still unassigned, it can assign time-slots for itself. Then the result is broadcasted to (H + 1)-hop away. Th3eprocess continues until all nodes are scheduled.

```
Algorithm Distributed ONPC Time-slot Assignment
Initialize-Process( )
1. Initialized its own state as UNASSIGNED, and broadcast a HELLO
   message ($H$ + 1)-hops away with its state, id and node degree.
2. Build the neighbor table NeighborTable with incoming HELLO
   message.

Assign-Process()
1. Compare node degree and node id with the nodes in the NeighborTable
   whose state is UNASSIGNED.
2. If it is not the one with largest node degree and node id, wait until
   receiving an ASSIGNMENT message originated from one node, modify
   the state and assign-result for this originated node in the NeighborTable,
   return 1.
3. Select the least number of time-slot that can be used to transmit to all
   one-hop neighbor nodes and satisfies conditions $C1$ ~ $C5$.
4. Change the node state as ASSIGNED. And broadcast an ASSIGNMENT
   message out to ($H$ + 1)-hops away.
```

Fig. 4. The distributed ONPC algorithm

```
Algorithm Distributed ANPC Time-slot Assignment
Initialize-Process( )
1. Initialized its own state as UNASSIGNED, and broadcast a HELLO
   message ($H$ + 1) hops away with its state, id and node degree.
2. Build the neighbor table NeighborTable and NFixed with the incoming
   HELLO message.

Assign-Process()
1. Compare node degree and node id with the nodes in the NeighborTable
   whose state is UNASSIGNED.
2. If it is not the one with largest node degree and node id, wait until
   receiving an ASSIGNMENT message originated from one node,
   modify the state and assign-result for this originated node in the
   NeighborTable, return 1.
3. For each channel $k$ in NFixed($u$), select the least number of time-slot
   that can be used by node $u$ to transmit to all neighbors whose fixed
   channel is $k$, and satisfy conditions $C1$ ~ $C5$.
4. Change $u$'s state as ASSIGNED. And broadcast an ASSIGNMENT
   message out to ($H$ + 1) hops away.
```

Fig. 5. The distributed ANPC algorithm

Accordingly each node maintains a table as *NeighborTable* with five fields, namely, *node-id*, *node-degree*, *fixed-channel*, *state* and *assign-results*. The *node-id* and *node-degree* denote the node identification and the number of one-hop neighbors; the *fixed-channel* is the number of the fixed channel, and the *assign-results* are the scheduling results for this node.

The distributed versions of the ONPC and ANPC are listed in Figure 4 and 5. There are two processes, the Initialize-Process and Assign-Process. In the Initialize-Process the node first initializes its state as UNASSIGNED, collects its *node-id*, *node-degree* and *fixed-channel* and broadcasts a HELLO message to (H + 1)-hop away. Finally each node builds its *NeighborTable*. In the Assign-Process the node compares its node degree with nodes in *NeighborTable* whose state is UNASSIGNED. If its node degree is the maximal, the node assigns time-slots for itself according to the *assign-results* in *NeighborTable* and broadcasts an ASSIGNMENT message to (H + 1)-hop away.

The differences between the distributed ONPC algorithm and ANPC algorithm are that the latter also builds the set of fixed channels of one-hop neighbors, i.e., *NFixed*, and assigns time-slots for each channel in *NFixed*.

Theorem 5. Each node transmits no more than $\frac{2K^{H+1}-2}{K-1}$ messages during the ONPC and ANPC algorithm.

Proof. During the Initialize-Process, each node sends once its list of neighbors and also forwards messages from neighbors within (H + 1)-hop away. The total number of messages transmitted in this process is no more than:

$$1 + K + K^2 + \ldots + K^H = \frac{K^{H+1}-1}{K-1}.$$

The Assign-Process is similar and the theorem stands. □

Theorem 6. The computation time complexity of the distributed ONPC and ANPC algorithms is $O(n)$.

Proof. By noticing that each node sends out a constant number of messages and a constant number of steps is performed per message. □

6 Simulation Results

The simulation is built in a 1000m× 1000m plane area with node number n as 50 or 150. The nodes are placed randomly and with same transmission range varing from 130m to 250m. The available channels Q is assumed to be 12 or 3. The fixed channel for each node is randomly selected among all available channels. For a given node number and transmission range, 1000 scenarios are built. The results are the average values of 1000 scenarios. The length of the scheduling cycle Δ is relative to the topology of the network. A new parameter *ratio* is introduced here as:

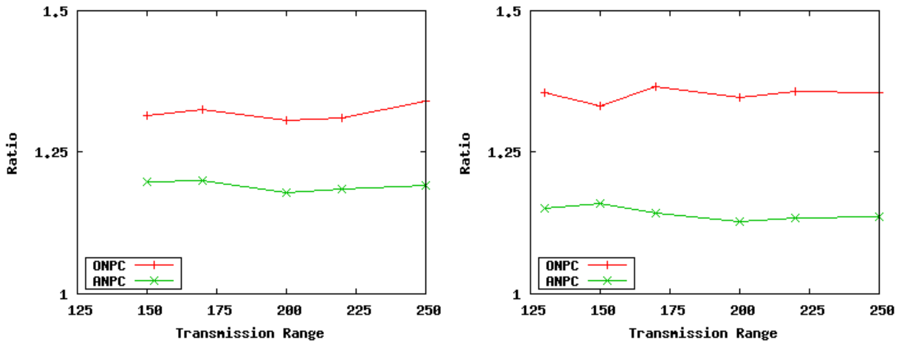
$$ratio = \frac{\Delta}{K}.$$

In which K represents the maximum node degree in the network.

Since most of the current works are based on the assumption that $H = 1$, we test the both cases $H = 1$ and $H = 2$ in the simulation. Figure 6 and 7 show the results with $H = 1$. The performance of ANPC is better than ONPC in all scenarios.

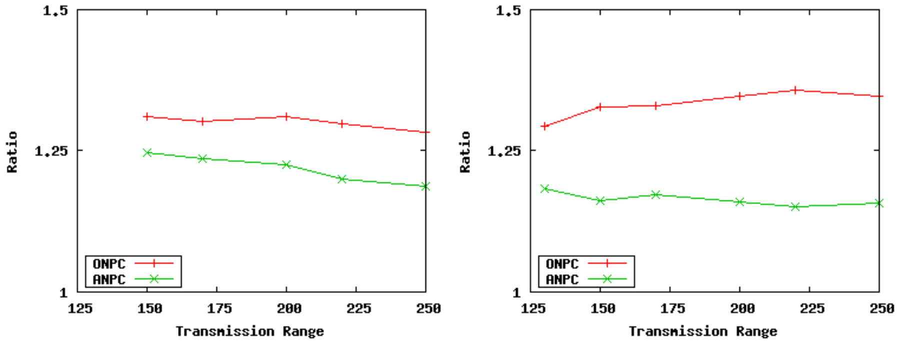
Although the length of the scheduling cycle is upper-bounded by the maximum number of $(H + 1)$-hop neighbors in the theoretic analysis. We can observe from the simulation results that the value of *ratio* is rather close to 1. It means that the average length of the scheduling cycle is close to the maximum node degree in the network. The results imply that a proper way to reduce the length of the cycle is to control the maximum node degree in the network through topology control methods, and thus the performance of the network can be improved significantly.

Figure 8 and 9 show the results in case that $H = 2$. The performance of the ANPC is better than the ONPC in all cases too. The length of the scheduling cycle increases



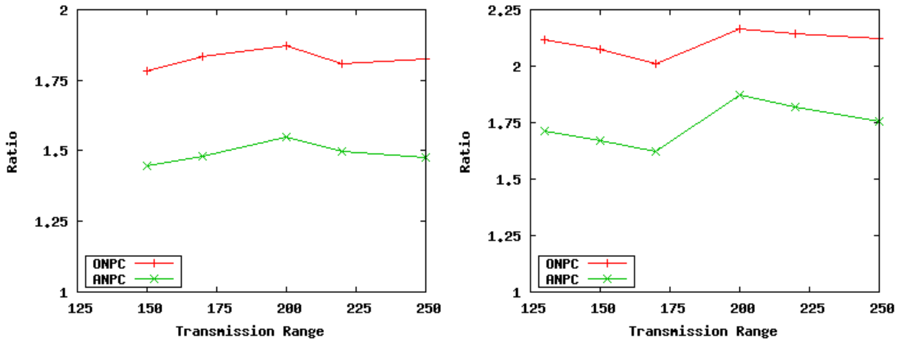
(a) $n = 50$. (b) $n = 150$.

Fig. 6. $H = 1$, $Q = 12$



(a) $n = 50$. (b) $n = 150$.

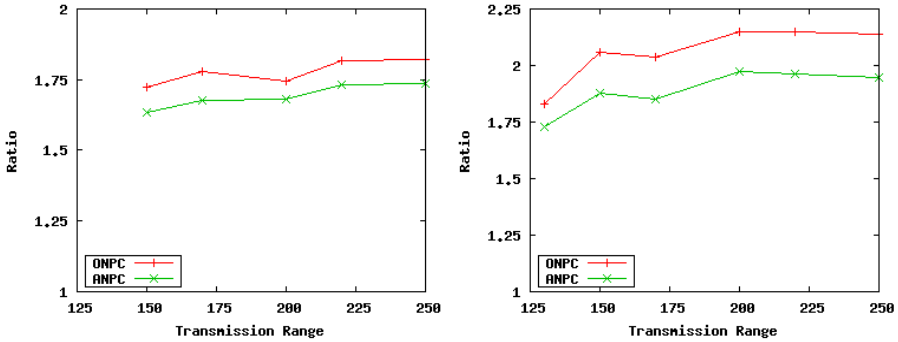
Fig. 7. $H = 1$, $Q = 3$



(a) $n = 50$.

(b) $n = 150$.

Fig. 8. $H = 2$, $Q = 12$



(a) $n = 50$.

(b) $n = 150$.

Fig. 9. $H = 2$, $Q = 3$

significantly compared with the case $H = 1$ and it will weaken the system performance. A compromise may be necessary to balance the QoS requirement and system utilization in the mesh networks which can be studied further.

7 Conclusions

In this paper we have studied the access scheduling problem on the control channels in the TDMA wireless mesh networks. The problem is to assign time-slots for each node in the network to access the control channels and the objective is to minimize the length of the scheduling cycle. Here we have shown the NP-completeness of the scheduling problem and proposed two heuristic algorithms for it, namely, One Neighbor Per Cycle (ONPC) and All Neighbor Per Cycle (ANPC). We also prove that the number of time-slots by the second algorithm is upper-bounded by $\min(n, 4K - 2)$ in some cases. The simplicity and locality make these algorithms rather suitable for wireless mesh networks. Simulation results also show that the performance of ANPC algorithm is much better.

References

1. Akyildiz, I., Wang, X., Wang, W.: Wireless Mesh Networks: a Survey. Computer Networks~47, 445--487 (2005)
2. Chlamtac, I., Pinter, S.: Distributed Nodes Organization Algorithm for Channel Access in a Multihop Dynamic Radio Network. IEEE Transactions on Computers~C-36(6), 729--737 (1987)
3. Cheng, H., Jia, X.: Distributed Channel Access Scheduling on the Control Channel in Wireless Mesh Networks (submitted to the WiCOM 2007) (2007)
4. Kyasanur, P., Chereddi, C., Vaidya, N.: Net-X: System eXtensions for Supporting Multiple Channels, Multiple Radios, and Other Radio Capabilities, Technical Report, Department of Computer Science, University of Illinois at Urbana-Champaign (2006)
5. Ramanathan, S., Lloyd, E.: Scheduling Algorithms for Multi-hop Radio Networks. IEEE/ACM Transactions on Networking, 166--177 (1993)
6. Xu, K., Gerla, M., Bae, S.: How Effective is the IEEE 802.11 RTS/CTS Handshake in Ad Hoc Networks. In: Proc. IEEE GLOBECOM 2002, Taipei, Taiwan, pp. 72--76 (November 2002)
7. Cidon, I., Sidi, M.: Distributed Assignment Algorithm for Multihop Packet Radio Networks. IEEE Transactions on Computers~38(10), 1353--1361 (1989)
8. Zhu, C., Corson, M.: A Five-Phase Reservation Protocol (FPRP) for Mobile Ad Hoc Networks. Wireless Networks~7, 371--384 (2001)
9. Tang, Z., Garcia-Luna-Aceves, J.: A Protocol for Topology-Dependent Transmission Scheduling in Wireless Networks. In: Proc. IEEE Communications and Networking Conference, vol.~3, pp. 1333--1337 (1999)
10. Cai, Z., Lu, M., Georghiades, C.: Topology-transparent Time Division Multiple Access Broadcast Scheduling in Multihop Packet Radio networks. IEEE Transactions on Vehicular~52(4), 970--984 (2003)
11. Su, Y., Su, S., Li, J.: Topology-transparent Link Activation Scheduling Schemes for Multihop CDMA Ad Hoc Networks. In: Proc. IEEE GLOBECOM 2004, vol.~6, pp. 3563--3567 (2004)
12. Ju, J., Li, V.: TDMA Scheduling Design of Multihop Packet Radio Networks Based on Latin Squares. IEEE Journal on Selected Areas in Communications~17(8), 1345--1352 (1999)
13. Chen, J., Ting, P., Lin, C., Chen, J.: A Novel Broadcast Scheduling Strategy Using Factor Graphs and Sum-product Algorithm. In: Proc. GLOBECOM 2004, vol.~6, pp. 4048--4053 (2004)
14. Lloyd, E.: Broadcast Scheduling for TDMA in Wireless Multi-Hop Networks. In: Stojmenovic, I. (ed.) Handbook of Wireless Networks and Mobile Computing, pp. 347--370. John Wiley and Sons Inc., England (2002)
15. Ngo, C., Li, V.: Centralized Broadcast Scheduling in Packet Radio Networks via Genetic-fix Algorithms. IEEE Transactions on Communications~51, 1439--1441 (2003)
16. Maxim 2.4 GHz 802.11b Zero-IF Transceivers, \url{http://pdfserv.maxim-ic.com/en/ds/MAX2820-MAX2821.pdf}
17. Chandra, R., Bahl, P.: MultiNet: Connecting to Multiple IEEE 802.11 Networks Using a Single Wireless Card. In: IEEE INFOCOM (2004)

An Energy Efficient Communication Protocol Based on Data Equilibrium in Mobile Wireless Sensor Network*

Shuai Gao, Yanchao Niu, Hongwei Huo, and Hongke Zhang

Beijing Jiaotong University, Beijing 100044, China
shgao@bjtu.edu.cn, gniux819@gmail.com, {hwhuo, hkzhang}@bjtu.edu.cn

Abstract. This paper considers large-scale dense wireless sensor networks with path-constrained mobile sink. From the viewpoints of improving data equilibrium and energy-saving, we propose an enhanced energy efficient communication protocol in which a partition algorithm of overlapping time based on data equilibrium is designed to minimize the data variance and disequilibrium from each node. We present a data equilibrium information diffusion method to control the amount of collected data to avoid the waste of data and energy. Simulations under OMNET++ and MATLAB show that with protocol and algorithms proposed in this paper we can achieve better performance than other conventional methods in terms of data equilibrium rate, network lifetime and energy dissipation equilibrium rate.

1 Introduction

Generally, mobility in traditional networks is caused by people or mobile carriers which move randomly and uncontrollably. Mobility is regarded as extra overhead and will bring more complexities and costs for communication system while providing more flexible applications and access approaches for users. In wireless sensor network (WSN), sensor nodes' positions can be changed not only passively by people, animals or vehicles, but also actively by themselves. Mobility may improve wireless sensor network performance [1,2,5,6,7,8], including prolonging network lifetime, increasing total network capacity, reducing energy consumption and so on with the aid of specific mobility algorithms and policies.

Various types of mobility approaches in WSN have been proposed which have respective focuses. In [1], random mobility of nodes is incorporated with ad hoc wireless networks to improve throughput performance. In [2,3], the authors propose an architecture with MULEs (Mobile Ubiquitous LAN Extensions) to collect sensor data in sparse WSN. The MULEs are acted by randomly moving humans and animals whose motion cannot be predicted in advance. With shorter range communication, this approach can lead to substantial energy saving. But

* This research is supported by the National Science Foundation of China (No.60572037).

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 433–444, 2007.

it's difficult to bound the data transfer latency in this approach due to random mobility, similar to [1]. The method proposed in [4] aims at changing locations of mobile sinks for energy minimization using ILP model. Differently from ILP formulation in [4], the objective function in [5] concerns the overall network time directly, instead of deducing it from energy consumption indirectly. The technique in [6] leads to a more uniform distribution of energy consumption based on the method of joint mobility and routing. In [4,5,6], sinks may choose mobile objective positions arbitrarily in total region regardless of the data transferring delay.

The certain application in this paper is about large-scale dense wireless sensor networks with path-constrained mobile sink. [9,10] is the first paper about path-constrained mobile WSN in which sensor nodes are deployed on buildings and data collection nodes called mobile observer are placed on transportation vehicles which mobile trajectory and velocity are fixed and predictable. Sensor nodes only directly transmit data when the mobile observer is close to them. A queuing model is introduced to model the data collection process for mathematical and systematic analysis. The approach using predictable mobility can effectively save energy with boundedness of transfer delay due to the determinism of the path and time. Similarly, the mobile trajectory is constrained in [11]. Which are different is that the sink's velocity is adaptable and the sensor nodes transmit data to mobile sink with multi-hop relaying. [11] proposes an adaptive mobile policy and communication protocol to increase the number of packets transmitted and reduce energy consumption.

By contrast with static WSN, mobile WSN can solve the "energy hot spot" problem effectively by the mobility of sink or sensor nodes. But energy-saving in mobile wireless sensor network is an ever-lasting focus and very important due to the scarce energy resource of WSN. Aiming at the path constrained large-scale dense WSN, this paper proposes an enhanced energy efficient communication protocol based on equilibrium of data collected, including a mobile hand-off algorithm based on minimized data variance and data equilibrium information diffusion method to reduce energy consumption and enhance network lifetime.

The remainder of this paper is organized as follows. In Section 2, we analyze the application background systematically and pose two important factors which should be considered. An enhanced energy-saving communication protocol is proposed in Section 3. Section 4 is devoted to the minimized data variance algorithm within identical overlapping time group. The system simulation and evaluation are presented in Section 5. Finally, Section 6 summarizes this paper and discusses the future work.

2 Application Background Analysis

This paper focuses on large-scale dense wireless sensor networks with path-constrained mobile sink which specific applications include ecological environment monitoring, health monitoring of large buildings and so on. Figure 1 shows

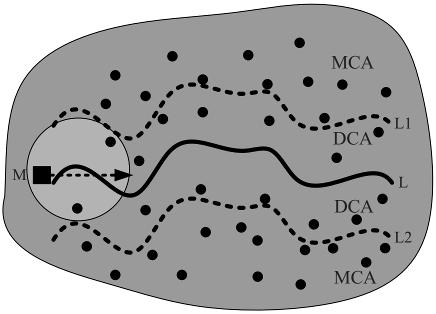


Fig. 1. An example of path-constrained mobile WSN

an abstract example of this kind of applications in which mobile sink installed in robots or cars moves periodically along existing road infrastructure with a result that the trajectory of mobile sink is fixed and unchanged. Sensor nodes are deployed around the mobile trajectory L. Mobile sink M communicates with sensor nodes when moving closely to them. According to the communication range of M, the monitored region can be divided into two parts, direct communication area (DCA) and multi-hop communication area (MCA). In Figure 1, the area between L1 and L2 is DCA in which sensor nodes can directly transmit data to M due to closer distance to mobile trajectory. For sensor nodes in MCA, multi-hop relaying is necessary. [11] proposes a multi-hop clustering algorithm in which nodes of DCA are chosen as cluster heads and nodes of MCA join into the nearest clusters. The data from nodes in MCA is forwarded to their own cluster head which transmits the total data of the cluster to mobile sink finally. After every movement round, mobile sink may send the collected information to database for data process. This kind of specific applications has the following properties:

- Mobile sink moves periodically with fixed trajectory and velocity.
- Mobile sink may communicate with no more than one sensor node in DCA anytime.
- Mobile sink has unlimited energy, memory and computing resource.
- This application is delay-tolerant.

There are two factors that must be considered in the above kind of WSN.

Firstly, the total amount of data transmitted from cluster head to mobile sink in one movement round is limited due to low data rate which is determined by low-energy wireless MAC and PHY techniques. It's possible for one cluster head not to transfer all data colleted to mobile sink that will result in waste of data and energy. Controlling the amount of data collected effectively is necessary to reduce the energy consumption.

Secondly, sensor nodes are deployed randomly which brings complex problems. The number of every cluster and the communication time between mobile

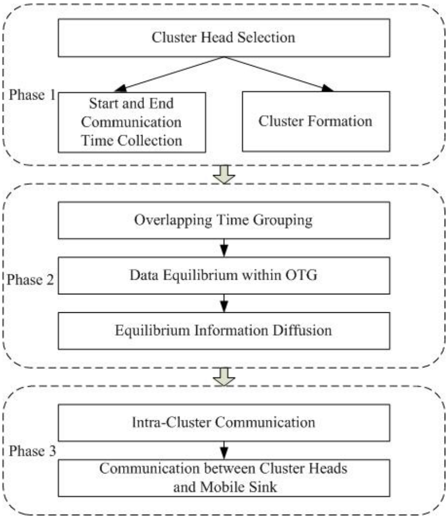

Phase 1
Cluster Head Selection
Start and End Communication Time Collection
Cluster Formation
Phase 2
Overlapping Time Grouping
Data Equilibrium within OTG
Equilibrium Information Diffusion
Phase 3
Intra-Cluster Communication
Communication between Cluster Heads and Mobile Sink

Fig. 2. Enhanced energy efficient communication protocol

sink and cluster heads can't be predicted. Large-scale cluster may communicate with short data transferring time while small-scale cluster with rather long data transferring time. It will bring about severe data disequilibrium between sensor nodes and difficulties to evaluate and analyze the concerned data effectively.

Based on the above factors, this paper is devoted to addressing energy saving problem from the viewpoint of balancing the data collected.

3 Enhanced Energy Efficient Communication Protocol

The mobile WSN communication protocols proposed in [10,11] don't consider the balance of the amount of data collected. This paper designs a new enhanced energy efficient communication protocol on the base of [10,11] which consists of three phases: (1)cluster formation, (2)data equilibrium information diffusion and (3)sensor data transferring, as shown in Figure 2. In phase 1 and phase 2, mobile sink move along fixed trajectory for a round respectively. Mobile sink begins to collect data from cluster heads from phase 3.

By contrast with the communication protocol in [11], phase 2 is a new enhanced function with optimized communication handoff method and data equilibrium information diffusion. And the cluster formation algorithm in phase 1 and data communication mechanisms in phase 3 may adopt the related techniques of [11]. Otherwise, mobile sink will collect the information of start time and end time when each cluster head enters into and departs from the communication range of mobile sink which is useful for overlapping time grouping. The following will describe and analyze the details of phase 2.

3.1 Overlapping Time Grouping

Let N_i denote the i_{th} cluster head which enters into the communication range of mobile sink. After phase 1, the physical information about the monitored region colleted by mobile sink includes:

- T_i^{start}: start time when mobile sink enters into N_i's communication range
- T_i^{end}: end time when mobile sink leaves from N_i's communication range
- H_i: the number of N_i's cluster members
- C_{head}: the number of cluster heads in monitored region
- N_{total}: the total number of sensor nodes in monitored region

If $T_{i+1}^{start} < T_i^{end}$, N_i and N_{i+1} simultaneously located in the communication range of mobile sink then it is called overlapping. It's possible that more than two nodes overlap when sensor nodes are deployed randomly. This paper will focus on no more than two nodes overlapping. For more complex overlapping, we will do in-depth research in future.

According to overlapping of N_i and adjacent cluster heads, total cluster heads can be divided into G overlapping time group (OTG) following Algorithm 1.

Algorithm 1. Overlapping time grouping algorithm

```
Initialize Number of OTG to 1
Initialize cluster heads number of the first group to 1
while check all cluster heads in sequence of entering into the communication range
      of mobile sink do
    if overlapping then
        Cluster heads number of current group increases by 1
    else
        Number of OTG increases by 1
        Point to new group and initialize cluster heads number of new group to 1
    end if
end while
```

3.2 Data Equilibrium Within Overlapping Time Group

After running Algorithm 1, all continuous overlapping cluster heads will be listed into identical group and no overlapping occurs between groups. Let G_j denote the j_{th} OTG and C_{G_j} denote the cluster heads number of G_j . According to the Algorithm 1, we can derive $\sum_{j=1}^{G} C_{G_j} = C_{head}$. If $C_{G_j} = 1$, then there's only one cluster head in this group. And the only cluster head may transmit its data to mobile sink with maximized time without handoff algorithm. If $C_{G_j} > 1$, it's necessary to run minimized variance algorithm (MinVar) of data amount within OTG to optimize the partition of overlapping time. See Section 4 for the details of MinVar algorithm.

When one cluster head communicate with mobile sink, the data is chosen in a round-robin fashion [11] within all sensor nodes administrated by the cluster head. The purpose of round-robin fashion is approximate data equality of each sensor node. Using MinVar algorithm, the variance of data from each sensor nodes within OTG can be minimized in round-robin fashion and the disequilibrium of sensed information in monitored region can be lowered effectively.

3.3 Diffusion of Data Amount Equilibrium Information

According to the energy model in [11], the energy consumed when transmitting and receiving is proportional to the amount of data. In [11], cluster head and its cluster members wake up to collect physical information when mobile sink enters into the communication range of this cluster head and turn to sleep mode from active mode when mobile sink leave from the cluster head. Let T_i be the data transferring time of cluster head N_i, then the total amount of data collected at a rate D_s by sensor nodes during T_i is $H_i * D_s * T_i$. And the maximum of information amount transmitted from N_i to mobile sink during T_i is $D_t * T_i$. Here, D_t is the data rate between cluster heads and mobile sink. If $D_t < H_i * D_s$, there are part of data collected that can't be transmitted to mobile sink leading to waste of energy.

Using MinVar algorithm in phase 2, we can get the maximum amount of information from each node under the condition of optimal data equilibrium within OTG. Then, we can diffuse this maximum information to the monitored region to control the data collected in each round. After the diffusion, sensor nodes can collect enough data for transmitting and then turn to sleep mode immediately. This method of diffusion is used to control the collected data amount, so we can call it Controlled Collection (CC). CC can obviously avoid the waste of collected data and reduce the system energy consumption. Correspondingly, the approach in [11] without controlling collection according network topology and nodes' deployment is called Uncontrolled Collection (UC).

4 Minimized Variance Algorithm of Data Amount (MinVar)

Let V_i denote the i_{th} cluster head in overlapping time group G_j, then the number of V_i's cluster members $H_i^{'} = H_{G_j.index+i-1}$. Here, $G_j.index$ is the sequence of the first cluster head of G_j in all monitored cluster heads. Figure 3 depicts the partition of overlapping time in G_j. The parameters definitions in Figure 3 are as follows.

- n: the number of cluster heads in G_j, $n = C_{G_j}$
- s_i: overlapping time length between V_i and V_{i+1}
- t_i: the time length when only V_i locates in mobile sink's communication range
- x_i: the time length allocated to V_i in overlapping time s_i

We can derive the total time T when mobile sink communicates with all cluster heads from the start time of the first node and the end time of the last node in G_j.

$$T = T^{end}_{G_j.index+C_{G_j}} - T^{start}_{G_j.index}$$

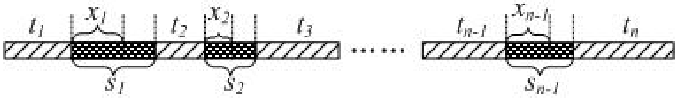


Fig. 3. Partition of overlapping time

Cluster heads transmit data at a rate D_t and the total amount of data between mobile sink and all cluster heads is $Q_{G_j} = T * D_t$. Then the mean of data amount from each sensor node is:

$$P_{G_j} = \frac{Q_{G_j}}{\sum_{l=G_j.index}^{G_j.index+C_{G_j}} H_l} = \frac{T * D_t}{\sum_{l=G_j.index}^{G_j.index+C_{G_j}} H_l} = P'_{G_j} * D_t$$

For each cluster head V_i, the data amount transmitted from V_i to mobile sink is:

$$Q_{V_i} = \begin{cases} (x_1 + t_1) * D_t & i = 1 \\ (x_i - x_{i-1} + t_i + s_{i-1}) * D_t & 1 < i < n \\ (t_n + s_{n-1} - x_{n-1}) * D_t & i = n \end{cases}$$

Then we can achieve the mean of data amount transmitted by each V_i's cluster member:

$$P_{V_i} = Q_{V_i}/H'_i = (t_i + x_i) * D_t/H'_i$$

The optimization goal of data equilibrium is: For overlapping time group G_j, seeking the best combination of $(x_1\ x_2\ x_3 \cdots x_{n-1})$ to minimize the variance of P_{V_i}. That's to design a optimal handoff algorithm to balance the data collected from each node.

The mathematical description of the above problem is:

$$min\ \ Var(P_{V_i})\ = min\ [E(P_{V_i} - P_{G_j})^2] = \ min\ [\sum_{i=1}^{n}(P_{V_i} - P_{G_j})^2]/n \qquad (1)$$

$$s.t.\ \ 0 \leq x_i \leq s_i$$

The partition of overlapping time is independent of n in the denominator of Equation 1. For simplicity, we will omit this parameter. $Var(P_{V_i})$ is a function with $(x_1\ x_2\ x_3 \cdots x_{n-1})$ as independent variables. So (1) has another expression as follows.

$$min\ \ F(X)\ = \sum_{i=1}^{n} f_i^2(X) \qquad (2)$$

$$f_i(X) = AX - B = p_i^T X - b_i \quad s.t.\ \ X - D \leq 0 \quad x_i \geq 0 \qquad (3)$$

$$X = (x_1,\ x_2,\ x_3, \cdots, x_{n-1})^T \quad D = (s_1,\ s_2,\ s_3, \cdots, s_{n-1})^T$$

From Equation 2 and Equation 3, it can be induced that the optimization function in Equation 1 is an kind of linear inequality-constraint least squares adjustment. The following is the details about matrix A and B in Equation 3.

$$A = \begin{pmatrix} p_1^T \\ p_2^T \\ p_3^T \\ .. \\ p_n^T \end{pmatrix} = \begin{pmatrix} 1/H_1^{'} & 0 & 0 & 0 & \cdots & 0 \\ -1/H_2^{'} & 1/H_2^{'} & 0 & 0 & \cdots & 0 \\ 0 & -1/H_3^{'} & 1/H_3^{'} & 0 & \cdots & 0 \\ \multicolumn{6}{c}{\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots} \\ 0 & 0 & 0 & \cdots 0 & -1/H_n^{'} \end{pmatrix} \times D_t \tag{4}$$

$$B = \begin{pmatrix} P_{G_j}^{'} - t_1/H_1^{'} \\ P_{G_j}^{'} - (t_2 + s_1)/H_2^{'} \\ P_{G_j}^{'} - (t_3 + s_2)/H_3^{'} \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ P_{G_j}^{'} - (t_n + s_{n-1})/H_n^{'} \end{pmatrix} \times D_t \tag{5}$$

Obviously, $A^T A$ is a nonsingular matrix. According to the rule of linear least squares [12], the optimization function in Equation 2 can be solved.

5 Simulation Results

We adopt OMNET++ [13] to validate our approach. In the experiments, the network consists of 60 uniformly placed sensor nodes in a 400m*160m area. Cluster heads transmit data to mobile sink at a rate 30KB/s with 2mw transmit power and receive power. The data rate sensor nodes collecting information is 10KB/s. And the maximum communication distance and initial energy is set to 52m and 10J respectively. Mobile sink moves along the short boundary of the rectangle area monitored with the speed of 5m/s. In this simulation environment, all cluster heads will overlap each other due to dense deployment.

We will validate the proposed communication protocol in term of data equilibrium rate, network lifetime and equilibrium rate of energy consumption.

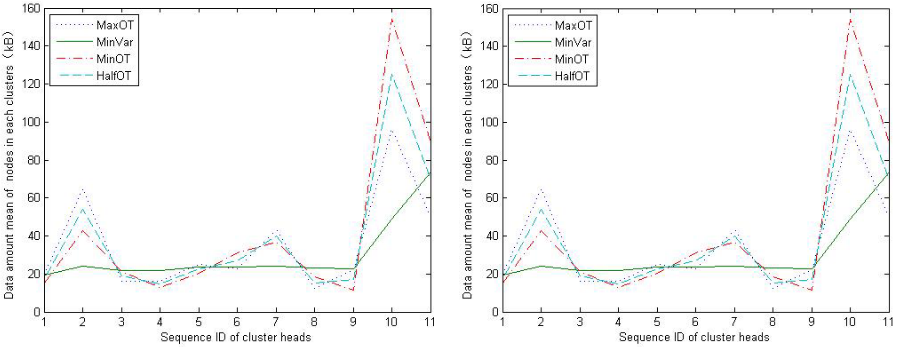
1) Data Equilibrium Rate (DER): equilibrium degree of data amount from each node within identical overlapping time group which can be expressed by the variance of data amount.

2) Network Lifetime: the number of movement round of mobile sink in which the first node' energy exhausts.

3) Equilibrium Rate of Energy Dissipation (ERED): it's similar with DER and can be reflected by the variance of each node's energy consumed or residual energy.

The performance of DER is compared between MinVar and three other following approaches.

1) Minimal Overlapping Time (MinOT): mobile sink will disconnect current communication link with current cluster head and turn to new cluster head once it detects new cluster head entering into its communication range.



(a) Data amount mean of sensor nodes in each cluster head per round

(b) Data amount variance of all under multiple random experiments

Fig. 4. Data equilibrium rate with four partition methods

2) Maximal Overlapping Time (MaxOT): mobile sink doesn't disconnect the communication link with current cluster head until it detects that the cluster head has left from its communication range. The partition method of the overlapping time in [11] adopts MaxOT.

3) Half Overlapping Time (HalfOT): overlapping time is divided into two parts evenly, i.e. $x_i = s_i/2$

Figure 4(a) displays the mean data amount of sensor nodes in a round of each cluster head with four overlapping time partition methods. It's clear that 11 nodes close to the mobile trajectory are chosen as cluster heads from X-coordinate. In Figure 4(a) the curves of data amount fluctuate drastically in other three methods but MinVar which produces in rather less changes. It can be concluded that MinVar algorithm has better DER than other three methods.

To prove the above conclusion about DER more strongly, we extend the experiments in Figure 4(a) to simulate data amount variance of all nodes under condition of different number of cluster heads and different partition methods (See Figure 4(b)). In the extended experiments, we repeat the random simulations 100 times in each different cluster heads number and compute the mean of multiple variances. Using MinVar algorithm can reduce the variance and improve the data rate equilibrium obviously and effectively.

Figure 5 and Figure 6 show the affect of different collection control methods on system total energy consumed per round and network lifetime. The term 'CC-MinVar' denotes controlling the amount of data collected using MinVar partition methods and other terms 'CC-XXX' are similar with 'CC-MinVar'. The total energy consumed per round is the sum of all nodes' energy consumption in receiving and transmitting data during a movement round of mobile sink.

As expected, 'CC-XXX' approaches are more energy efficient due to the control of data collected actively. Among the four 'CC-XXX' approaches, using CC-MinVar will obtain the least system total energy consumed per round.

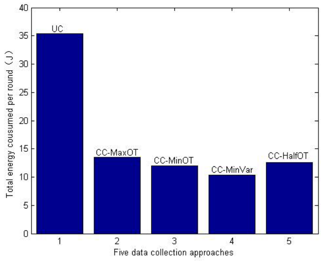


Fig. 5. Total energy consumed per round

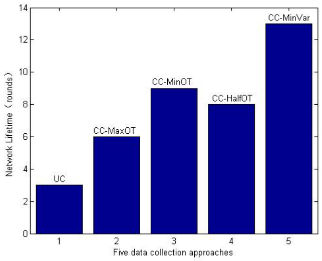
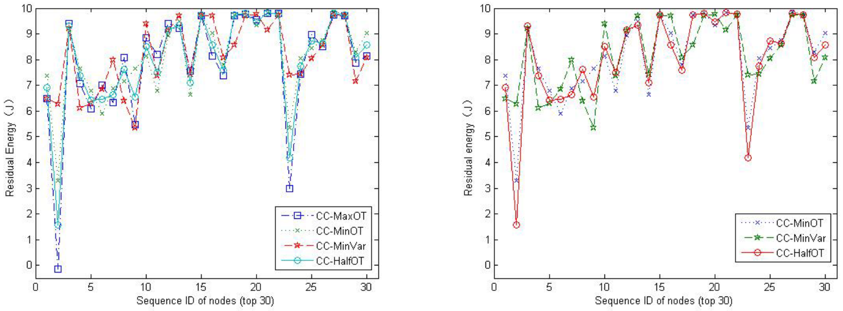


Fig. 6. Network Lifetime

The comparison of network lifetime in Figure 6 acts in accord with the energy consumption in Figure 5 CC-MinVar can bring the longest network lifetime and prolong the lifetime 433% than NCC. The reason why CC-MinVar is the best is due to higher equilibrium rate of energy consumption. MinVar algorithm brings about higher data equilibrium rate directly and higher energy consumption equilibrium rate indirectly after equilibrium information diffusion. Table 1 shows the variance of energy consumed per round of all nodes. In Table 1, MinVar has the least variance. The residual energy of top 30 sensor nodes after different rounds is given to performance the equilibrium rate of energy consumed animatedly in Figure 7. We can see that residual energy is more uniform in MinVar than

Table 1. Variance of energy consumed per round of each node

Approaches	CC-MaxOT	CC-MinOT	CC-MinVar	CC-HalfOT
Variance of energy	0.0864	0.0505	0.0376	0.0645



(a) after 6 rounds

(b) after 8 rounds

Fig. 7. Residual energy of sensor nodes after different rounds

other approaches in which partial nodes' energy dissipating very quickly results in shorter network lifetime.

6 Conclusion and Future Work

In this paper, we propose an enhanced energy efficient communication protocol aiming at the background of large scale dense wireless sensor network with path-constrained mobile sink. In the proposed protocol, a minimized variance algorithm of data mount is designed to partition the overlapping time. Then data equilibrium information is diffused to the monitored area to control the data amount collected by sensor nodes to avoid the waste of energy. Simulation experiments under OMNET++ and MATLAB shows that the protocol and algorithms presented in this paper outperform other conventional techniques in terms of data equilibrium rate, network lifetime and energy dissipation equilibrium rate. Using the proposed approaches, we can improve the equilibrium rate of collected data and energy consumed and prolong the network lifetime effectively.

We plan to focus our future work on, more complex overlapping application (three or more sensor nodes overlapping), energy efficient mobile clustering algorithms, communication protocol with multiple mobile sinks.

References

1. Grossglauser, M., Tse, D.N.C.: Mobility Increases the Capacity of Ad Hoc Wireless Networks. IEEE/ACM Trans. Networking(TON) 10(4) (August 2002)
2. Shah, R.C., Roy, S., Jain, S., Brunette, W.: Data MULEs: Modeling a Three-Tier Architecture for Sparse Sensor Networks. In: SNPA. Proc. IEEE Workshop Sensor Network Protocols and Applications (2003)
3. Jain, S., Shah, R.C., Borriello, G., Brunette, W., Roy, S.: Exploiting Mobility for Energy Efficient Data Collection in Sensor Networks. In: Proc. Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (2004)
4. Gandham, S.R., Dawande, M., Prakash, R., Venkatesan, S.: Energy efficient schemes for wireless sensor networks with multiple mobile base stations. In: proc. IEEE Globecom 2003, vol. 1 (December 1-5 , 2003)
5. Wang, Z.M., Basagni, S., Melachrinoudis, E., Petrioli, C.: Exploiting Sink Mobility for Maximizing Sensor Networks Lifetime. In: Proc. 38th International Conference on System Sciences (2005)
6. Luo, J., Hubaux, J.-P.: Joint Mobility and Routing for Lifetime Elongation in Wireless Sensor Networks. In: Proc. IEEE Infocom (2005)
7. Ekici, E., Gu, Y., Bozdag, D.: Mobility-Based Communication in Wireless Sensor Networks. IEEE Communications Magazine (July 2006)
8. Mergen, G., Zhao, Q., Tong, L.: Sensor Networks with Mobile Access:Energy and Capacity Considerations. IEEE Trans. Communication 54(11) (November 2006)
9. Chakrabarti, A., Sabharwal, A., Aazhang, B.: Using Predictable Observer Mobility for Power Efficient Design of Sensor Networks. In: Zhao, F., Guibas, L.J. (eds.) IPSN 2003. LNCS, vol. 2634, Springer, Heidelberg (2003)

10. Chakrabarti, A., Sabharwal, A., Aazhang, B.: Communication Power Optimization in a Sensor Network with a Path-Constrained Mobile Observer. ACM Transaction on Sensor Networks 2(3) (August 2006)
11. Somasundara, K., Jea, D., Estrin, D., Srivastava, M.: Controllably Mobile Infrastructure for Low Energy Embedded Networks. IEEE Trans. Mobile Computing 5(8) (August 2006)
12. Lawson, C.L., Hanson, R.J.: Solving Least Squares Problems[M]. Prentice-Hall, Inc., Englewood Cliffs, N.J (1974)
13. OMNET++3.3 (October 2006), http://www.omnetpp.org

Distributed Computation of Maximum Lifetime Spanning Subgraphs in Sensor Networks*

Harri Haanpää, André Schumacher, Thorn Thaler, and Pekka Orponen

Lab. for Theoretical Computer Science, TKK – Helsinki University of Technology,
P.O. Box 5400, FI-02015 TKK, Finland
Harri.Haanpaa@tkk.fi, Andre.Schumacher@tkk.fi,
Thorn.Thaler@tkk.fi, Pekka.Orponen@tkk.fi

Abstract. We present a simple and efficient distributed method for determining the transmission power assignment that maximises the lifetime of a data-gathering wireless sensor network with stationary nodes and static power assignments. Our algorithm determines the transmission power level inducing the maximum-lifetime spanning subgraph of a network by means of a distributed breadth-first search for minmax-power communication paths, i.e. paths that connect a given reference node to each of the other nodes so that the maximum transmission power required on any link of the path is minimised. The performance of the resulting Maximum Lifetime Spanner (MLS) protocol is validated in a number of simulated networking scenarios. In particular, we study the performance of the protocol in terms of the number of required control messages, and compare it to the performance of a recently proposed Distributed Min-Max Tree (DMMT) algorithm. For all network scenarios we consider, MLS outperforms DMMT significantly. We also discuss bringing down the message complexity of our algorithm by initialising it with the Relative Neighbourhood Graph (RNG) of a transmission graph rather than the full graph, and present an efficient distributed method for reducing a given transmission graph to its RNG.

1 Introduction

Maximising the lifetime of a network, most commonly in terms of connectivity, is a key design goal in wireless sensor networks. Network longevity can be affected by many methods, ranging from hardware design to energy-aware routing [1]. We focus on *topology control*, specifically on assigning transmission power levels to the battery-operated nodes so that under a uniform traffic load the network remains connected for a maximum length of time [2]. We consider the case where the nodes are non-mobile and the power levels, once fixed, stay the same throughout the operating life of the network. An application scenario would be a sensor network whose main purpose is to provide sporadic status messages to a common sink node. Consider for example a sensor network that is deployed in a forest region to detect fire.

* Partially supported by the Academy of Finland under grant 209300 (ACSENT).

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 445–456, 2007.

It is apparent that under our assumptions of stationary nodes and uniform traffic load, maximising the lifetime of a network is equivalent to finding the lowest possible transmission power levels for the nodes that suffice to make the network connected. This problem of minimising the maximum transmission power required to establish connectivity has been considered previously in the literature several times. One of the earliest papers on the topic is by Ramanathan and Rosales-Hain [3], who address the problem in the setting of maximising the lifetime of a single-session broadcast. They propose a centralised algorithm for finding the minimum maximum (minmax) transmission power level that maintains network connectivity, as well as two simple distributed heuristics that aim at achieving the same. Their distributed heuristics, however, are suboptimal and do not necessarily guarantee connectivity in all cases.

Kang and Poovendran [4] discuss several problems related to dynamic lifetime maximisation, such as non-uniform energy levels. They also emphasise the importance of considering the minmax energy metric rather than the more often addressed minimum total energy metric for maximising network lifetime. For a distributed implementation, Kang and Poovendran rely on distributed methods for constructing minimum spanning trees (MST), such as the algorithm of Gallager, Humblet and Spira [5]. These techniques are, however, rather involved, and we complement this work by suggesting a much simpler distributed method for constructing general spanning *subgraphs* with minmax edge costs.

The problem of minimising the *total*, as opposed to minmax, network transmission power required for connectivity has been studied extensively (cf. e.g. [2] and the references therein). Rodoplu and Meng [6] present a distributed algorithm for this problem that is based on the concept of *relay regions*: each node is aware of its own geographic location and the location of its neighbours. Based on a path-loss model, nodes can locally determine to which neighbour they should forward the message to minimise the total energy consumption. The algorithm proposed in [6] is optimal but requires extensive assumptions, such as the availability of location information and a specific path-loss model.

Wattenhofer, Li, Bahl, and Wang [7] propose a distributed algorithm for the same problem. Their algorithm, which relies on a geometric cone-based forwarding scheme, requires that nodes can measure exactly the direction of incoming radio transmissions (angle of arrival). It also makes further assumptions on geometric properties of the underlying graph model.

Furthermore, several researchers have proposed distributed algorithms that construct minimum spanning trees and can potentially be used for lifetime maximisation, e.g. the algorithm proposed in [5] or the self-stabilising algorithm by Gupta and Srimani [8]. An MST based problem formulation seems appropriate for minimising the total energy expenditure. However, the distributed construction of an MST is usually more involved than the distributed search for a spanning subgraph with minmax edge cost. In Section 3, we present a very simple and efficient distributed algorithm that finds such a spanning subgraph. For a discussion of the two different objectives, minimising total transmission power and minimising maximum transmission power, see e.g. [2,4].

Our Maximum Lifetime Spanner (MLS) protocol is based on an approach similar to the distributed MST algorithm of Gupta and Srimani [8], viz. the construction of paths with minmax edge cost by breadth-first search similar to the asynchronous Bellman-Ford algorithm. However, as we do not consider the construction of an MST, we obtain several simplifications of the resulting scheme. Furthermore, we observe that before running the algorithm we can prune the network according to an algebraic formulation of relative neighbourhood graphs (RNG), for the purpose of avoiding traversal of redundant edges.

Recently Guo, Yang, and Leung [9] proposed a distributed algorithm DMMT (Distributed Min-Max Tree) for constructing minmax edge cost multicast trees, in the style of Prim's MST algorithm. Since their technique can easily be adapted also to sensor network lifetime maximisation, and seems to be the proposal in the literature closest to our MLS approach, we conducted an experimental comparison of the runtime behaviour of the two algorithms DMMT and MLS.

The rest of the paper is organised as follows. Section 2 gives a formal description of the lifetime maximisation problem. Section 3 describes our distributed method for finding a spanning subgraph with minmax transmission cost in a given network. Section 4 discusses a generalisation of RNGs and how they can be utilised for improvements of our algorithm. Section 5 presents a distributed algorithm for finding the RNG as an initial step of the algorithm proposed in Section 3. In Section 6 we evaluate our proposed Maximum Lifetime Spanner algorithm in terms of the number of required control messages, and compare it to the performance of the Distributed Min-Max Tree algorithm [9] proposed by Guo et al. using the `ns2` network simulator [10]. Section 7 presents our conclusions and outlines future research directions.

2 Lifetime Maximisation and Optimal *p*-Spanners

We model a sensor network as a graph $G(\tau) = (V, E(\tau))$, where the set of vertices V corresponds to the nodes of the network, $\tau : V \mapsto \mathbb{R}^+$ is a transmission power assignment, and the set $E(\tau)$ represents the directed links between nodes induced by a given transmission power assignment τ. We assume distinct node identifiers.

Each node has a finite energy budget that is consumed during the operation of the network. We assume that the initial value is the same for all nodes. We consider a scenario where the energy consumed by wireless communication dominates over energy consumed by computation or sensing. The minimum power a node u can use to maintain a link to a neighbouring node v is denoted by $\delta(u, v)$, where $\delta : V \times V \mapsto \mathbb{R}^+$ is the representative link cost function. We assume that the link costs are symmetric, i.e. $\delta(u, v) = \delta(v, u)$ for all $u, v \in V$; this is the case for example if the costs represent signal attenuation resulting from a deterministic path-loss model that only depends on the pairwise distance of nodes. In practice, one would expect a number of unidirectional communication links between the nodes and choose the maximum of the edge costs for δ such that bidirectional communication can be supported. We consider the notion of lifetime that regards all nodes as equally important, so that the objective is to maximise the time span after which the first node runs out of energy [11].

The set $E(\tau)$ of edges in $G(\tau)$ is induced by the transmission power assignment τ by the rule that an edge (u, v) is an element of $E(\tau)$ if and only if the transmission power $\tau(u)$ at node u is at least $\delta(u, v)$. Each node has the same maximum transmission power $p_{\max}$ that must not be exceeded. We assume that the nodes can form a connected network by using full power, i.e., that $G(\tau_{\max})$ with $\tau_{\max}(u) = p_{\max}$ for all u is a connected graph.

We consider the problem of finding a static transmission power assignment $\tau : V \mapsto [0, p_{\max}]$ that maximises the lifetime of the network while retaining connectivity. In this context, the desired power assignment τ obviously induces a spanning subgraph with minmax edge cost α. Although this condition generally does not uniquely determine τ, choosing $\tau(u) = \alpha$ for all nodes u does not reduce the lifespan of the node that first runs out of energy. The power assignment τ is considered to be fixed after it has been once determined during the initial network setup. This property distinguishes this problem formulation from the computationally more complex problem of dynamically assigning transmission power levels [12].

Definition 1. *Given a set of nodes V and an edge cost function $\delta : V \times V \mapsto \mathbb{R}^+$, a graph $G = (V, E)$ is a p-spanner if G is connected and $\delta(u, v) \leq p$ for each edge $(u, v) \in E$. If no p'-spanner with $p' < p$ exists, then we say that the p-spanner is optimal.*

In other words, a p-spanner is a connected spanning graph for the nodes in V where no edge has cost greater than p. Note that for any network a $p_{\max}$-spanner exists exactly when the network can be connected by the nodes sending at full power. The lifetime maximisation problem is formulated as follows:

Definition 2. *Given a set V representing sensor network nodes and an edge cost function δ, find an optimal p-spanner $G = (V, E)$ for V and δ and determine a transmission power assignment $\tau : V \mapsto [0, p_{max}]$ such that $\max_{v \in V} \tau(v) \leq p$ and $\tau(u) \geq \delta(u, v)$ for each link $(u, v) \in E$.*

3 A Distributed Algorithm for Optimal *p*-Spanners

In the following, we describe a distributed algorithm that, given a graph G, finds a spanning subgraph of G with some minmax edge cost, i.e. an optimal p-spanner of G. Initially, we assume that each node v knows its neighbours in G and the cost between v and each of them. We assume that the links and costs are symmetric. Our algorithm finds a spanning subgraph – indeed, a spanning tree – with minmax edge cost as long as the original graph is connected. Following the reasoning presented in Section 2, the algorithm can then be used to determine the minmax transmission power that is required to maintain connectivity in a wireless sensor network. In this setting it would be run once during an initial setup phase of the network to distributively determine the transmission power

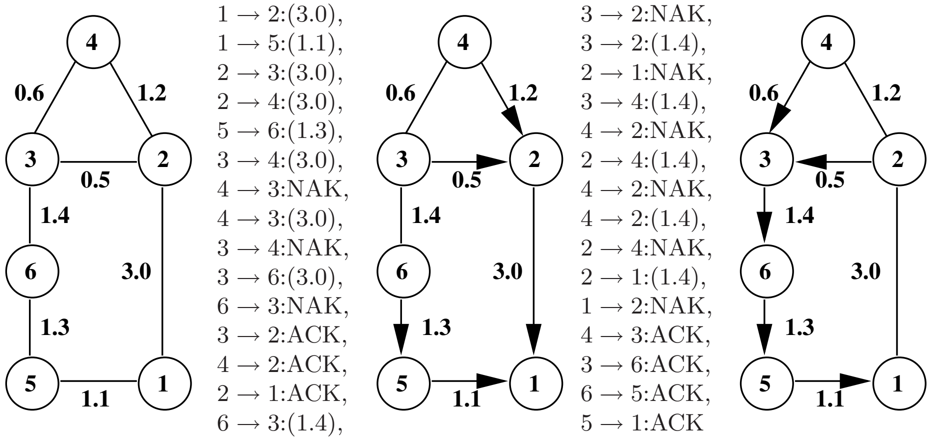
```
for node $v$ with local variables $\alpha, f, \alpha[\cdot], \text{status}[\cdot]$
at start:
    $\alpha \leftarrow \infty$; $f \leftarrow$ undefined
    for $u \in N(v)$:
        $\alpha[u] \leftarrow \infty$; status$[u] \leftarrow$ ready
    enter state SLEEP
in state SLEEP or state SEARCH:
    if $(\alpha')$ with $\alpha' < \alpha$ is received from some node $u$ then:
        if $f$ is defined: send NAK$(\alpha)$ to $f$
        $f \leftarrow u$
        for $w$ in $N(v) \setminus \{u\}$:
            if $\max(\alpha', \delta(v, w)) < \alpha[w]$:
                send $(\max(\alpha', \delta(v, w)))$ to $w$
                $\alpha[w] \leftarrow \max(\alpha', \delta(v, w))$; status$[w] \leftarrow$ wait
        enter state SEARCH
    if $(\alpha')$ with $\alpha' \geq \alpha$ is received from some node $u$ then: send NAK$(\alpha')$ to $u$
in state SEARCH:
    whenever status$[w]$=ready for all $w \in N(v) \setminus \{f\}$:
        send ACK$(\alpha)$ to $f$
        enter state SLEEP
    if ACK$(\alpha')$ or NAK$(\alpha')$ is received from $u$ and $\alpha[u] = \alpha'$ then: status$[u] \leftarrow$ ready
```

Algorithm 1. Distributed algorithm for finding an optimal p-spanner

level for each node. The graph $G(\tau_{\max})$ is then an obvious candidate for a graph to start from. Beneficial alternatives are discussed in Section 4 and 5.

Our Algorithm 1 for finding an optimal p-spanner is based on distributed breadth-first search similar to the asynchronous Bellman-Ford algorithm [13, Sec. 15.4]. However we use the properties of the minmax edge cost function to reduce the complexity of the search. First, a given reference node sends to each of its neighbours a message that contains the cost of the connecting edge. Upon first receiving the request, each node makes note of the node from which the message was received and rebroadcasts the request to its neighbours, updating the maximum edge cost α indicated in the request accordingly. Each node also remembers the best α sent to each neighbour. If a node that has already received and rebroadcast a request receives a request that indicates a better route from the reference node, it rebroadcasts the latter request to its neighbours if this leads to obtaining a route with a lower α, to those neighbours. In a typical data gathering scenario, the natural choice for the reference node is the node that collects the data.

Moreover, the nodes collect acknowledgements from their neighbours. When a node receives the request, it forwards it to its neighbours, and waits for each neighbour to either accept (ACK) or reject (NAK) it. When acknowledgements have been received from each neighbour, the node sends an ACK to the node from which it received the request. A NAK is sent if the node receiving the



1 → 2:(3.0),
1 → 5:(1.1),
2 → 3:(3.0),
2 → 4:(3.0),
5 → 6:(1.3),
3 → 4:(3.0),
4 → 3:NAK,
4 → 3:(3.0),
3 → 4:NAK,
3 → 6:(3.0),
6 → 3:NAK,
3 → 2:ACK,
4 → 2:ACK,
2 → 1:ACK,
6 → 3:(1.4),

3 → 2:NAK,
3 → 2:(1.4),
2 → 1:NAK,
3 → 4:(1.4),
4 → 2:NAK,
2 → 4:(1.4),
4 → 2:NAK,
4 → 2:(1.4),
2 → 4:NAK,
2 → 1:(1.4),
1 → 2:NAK,
4 → 3:ACK,
3 → 6:ACK,
6 → 5:ACK,
5 → 1:ACK

Fig. 1. Sample execution of Algorithm 1 from reference node 1; messages listed as *source→destination:message*. Initial state, intermediate state and final state with messages listed between states.

request already knows of a better path, or if a node learns of a better path while waiting for the acknowledgements from its neighbours. In this way, an ACK response means that the responding node has accepted the other node as its father in the tree being constructed, while a NAK signifies refusal. It can happen that a node will first respond with an ACK but later send a NAK; however, when the reference node has received acknowledgements from its neighbours, the algorithm has finished. To notify the remaining nodes about the termination of the algorithm, the reference node can then initiate a network-wide broadcast using the edges of the computed spanning tree. Each node v receiving this broadcast message can then decrease its transmission power $\tau(v)$ to the minimum power required to reach its father f_v and the neighbouring nodes that have chosen v to be their father. A sample run of Algorithm 1 is given in Figure 1.

In Algorithm 1, α is the current estimate of the minmax cost of a path from the reference node to each node v; and f is the node from which v has received the last accepted message. Initially, f is undefined and $\alpha = \infty$ for each v. The optimal p-spanner is defined by the f variables of each node after the algorithm has terminated.

To justify the algorithm, we firstly observe that it always terminates. Let Δ be the number of distinct edge costs in the graph; no node can learn of a new route with better α more than Δ times.

Secondly, at the end each node has a correct α_v: if from some node v there would exist a path of max cost $\alpha_0 < \alpha_v$ to the reference node, then on the path there is some edge of cost at most α_0 where exactly one endpoint would have a maximum edge cost estimate higher than α_0. This cannot happen, since the endpoint with cost at most α_0 should send a message along that edge. Thirdly, it cannot happen that a node would remain in the wait state, since its neighbours will respond to the queries either by an immediate NAK, if the cost was too large, a delayed ACK once the neighbour has received responses from its children, or a delayed NAK in case the neighbour later learns of a lower max cost path.

To consider the communication complexity of the algorithm, observe that the number of distinct edge costs is bounded by $\varDelta \leq |E|$ and with practical radio equipment, the number of distinct power levels is typically not large. In this regard the minmax edge cost spanner problem is different from finding minimum cost routes, where the number of routes with different total cost between two nodes can be exponential in the number of nodes [13, Sec. 15.4]. When a node learns of a better α, it will send a message to its neighbours, who will eventually answer with an ACK or a NAK. Since the requests sent by a node to its neighbour are in order of decreasing α, each of the $|E|$ edges participates in at most $2\varDelta$ updates, and the total communication complexity is $O(\varDelta\,|E|)$.

4 Relative Neighbourhood Graphs

Algorithm 1 requires nodes to exchange messages with all neighbours. In a dense sensor network where the number of nodes within transmission range may be large, it is beneficial to limit the number of nodes that need to be contacted, while maintaining network connectivity at the same minmax transmission cost. For this purpose, we use *relative neighbourhood graphs* [14]. Relative neighbourhood graphs and related structures have been used for topology control [15,16], mostly in a geometric context, where nodes are placed in a plane and $\delta(u, v)$ depends only on the Euclidean distance between u and v. However, we only assume that path loss is symmetric, i.e. $\delta(u, v) = \delta(v, u)$. We will find, though, that when the nodes are placed in the Euclidean plane, our algorithm runs much faster.

Definition 3. *Given a graph $G = (V, E)$ and an edge cost function δ, the relative neighbourhood graph of G is the graph with vertex set V and edge set $\{\{u, v\} \mid \{u, v\} \in E, \nexists w$ s.t. $\{u, w\}, \{w, v\} \in E, \delta(u, w) < \delta(u, v), \delta(w, v) < \delta(u, v)\}$.*

In effect, the relative neighbourhood graph is obtained by deleting from each triangle in the original graph the edge with the highest cost. Such a generalisation of the concept of RNG has been already successfully applied to other problems, such as searching and broadcasting in peer-to-peer networks [17].

Proposition 1. *For any p, the* RNG *of G contains a p-spanner if G does.*

Proof. Consider an optimal p-spanner in the original graph. Order the k edges removed from the original graph in constructing the relative neighbourhood graph in increasing order of cost as $e_1, e_2, \ldots, e_k$. Let E_0 denote the edge set of the RNG, and let $E_i = E_{i-1} \cup \{e_i\}$ for $0 < i \leq k$. Suppose for contradiction that E_0 admits no p-spanner. Since E_k admits a p-spanner, there must be some least $0 < i^* \leq k$ such that E_{i^*} admits a p-spanner. By definition of the RNG, in E_{i^*-1} the endpoints of e_i are connected by a path of two edges shorter than e_i, so E_{i^*-1} also admits a p-spanner – a contradiction.

5 Distributed Algorithms for RNGs

In this section, we describe a distributed method for constructing RNGs. We do not assume that a node initially knows about the cost of the edges to its neighbours, but we assume that a node can estimate the strength of arriving radio signals, e.g. using Received Signal Strength Indication (RSSI) for a system with IEEE 802.11 network interfaces.

To construct the RNG, each node beacons at maximum power, sending out a message that contains its distinct node identifier. Nodes learn about their neighbours by receiving beaconing messages from them. They also estimate the path loss from the received signal strength. Path loss is used to estimate the δ-cost of a particular edge. We assume that path loss is symmetric; e.g. all path loss functions where the path loss depends only on the distance between the nodes fall into this category. In this manner all nodes can learn about their neighbours in $O(|V|)$ beaconed messages of $O(1)$ size.

After having learned about their neighbours, the nodes prune unnecessary edges from the graph formed by the nodes and radio links. To this end, the nodes again send beaconing messages. In addition to the identity of the beaconing node, this time the messages also contain the list of neighbours of the beaconing node, and the associated δ costs. If a node u learns, upon receiving a message from node v, that for some third node w it holds that $\delta(u, w) > \delta(u, v)$ and $\delta(u, w) > \delta(v, w)$, then u can determine that the edge (u, w) is not in the RNG, as per Definition 3. Thus the nodes can prune their neighbourhood so that only the RNG remains in $O(|V|)$ messages, the size of each of which is proportional to the number of neighbours the beaconing node has, and $O(|E|)$ in total.

Pruning the connection graph down to the RNG before running Algorithm 1 can give very considerable savings in complexity. With an arbitrary path loss function, the RNG can still contain $O(|V|^2)$ edges. However, when the nodes are in a plane and path loss is an increasing function of distance, the RNG is a subgraph of the Delaunay triangulation of the original graph and contains only $O(|V|)$ edges [14]. Thus in the Euclidean plane the communication complexity of the entire algorithm, including beaconing to determine neighbours, determining the RNG and computing an optimal p-spanner is $O(|E| + \Delta |V|)$.

6 Simulations

We experimentally validated Algorithm 1 and compared its runtime behaviour to the Distributed Min-Max Tree (DMMT) algorithm by Guo, Yang, and Leung [9] for a number of different scenarios using the `ns2` network simulator.

6.1 The DMMT Algorithm

Although DMMT was recently proposed for constructing maximum lifetime multicast trees in wireless ad hoc networks, it can be readily applied to solve the lifetime maximisation problem as formulated in Section 2. We focus on the version of DMMT proposed in [9] for omnidirectional antennas. DMMT is based on Prim's well-known MST algorithm. The idea is to grow a subtree starting from the reference node, such that in each step the minimum cost edge is added that connects one node in the tree and another node not yet in the tree. After all nodes are added, the MST results.

The DMMT algorithm finds an optimal p-spanner (that is a tree) by adding an additional step to each iteration: after the minimum outgoing-edge-cost has been found, it is propagated to all tree nodes contained in a *join request* message. The tree nodes then forward this message to all adjacent nodes that are not yet in the tree using edges of cost at most the minimum outgoing edge-cost. Each node only forwards a join request once, and requests are identified by iteration counters. After a non-tree node is added via an edge incident to the tree node, the tree node becomes the *parent* of the added node which becomes a *child* of its parent. This is called the *growth phase*. After the growth phase has terminated, the next minimum outgoing edge-cost is determined in the subsequent *search phase*. In the search phase, each leaf node initiates a *join reply* message, which is forwarded along the tree towards the reference node. This message contains an estimate of the minimum outgoing-edge-cost in this iteration, which is updated as the message proceeds along the edges of the tree: each intermediate non-leaf node waits for all its children to send a *join reply* and then forwards a single *join reply* to its parent, that contains the minimum cost of the replies received from its children and the cost of its incident edges to non-tree nodes.

Guo et al. [9] have each node use timers to estimate the termination of the growth phase. However, to make DMMT more resilient against packet drops, we considered a more synchronised method where the source commands the nodes to switch from the growth to the search phase. The additional control messages required by our modification were not taken into account in comparison to MLS.

6.2 Experimental Evaluation of MLS

In evaluating MLS, we consider the number of control messages and the simulation time required. In our simulations, we use the *disk graph model*: the networks are created by randomly scattering nodes onto a square area with given dimensions, and connectivity is defined by the `ns2` default maximum transmission range. We discard disconnected graphs. Simulation parameters are summarised in Table 1. The dimensions of the square area were chosen to yield an expected density of one node per square of side length 130 m.

Table 1. Simulation parameters

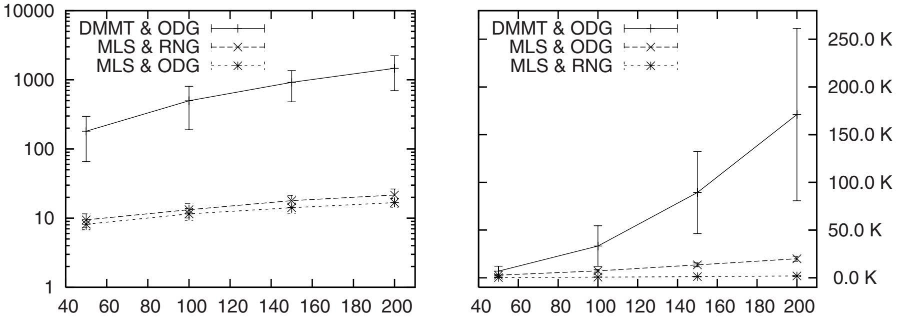
`ns2` version:	2.31	Square dimension:	919 m×919 m,1300 m×1300 m, 1592 m×1592 m,1838 m×1838 m
Transmission range:	250 m	Number of nodes:	50, 100, 150, 200
Interference range:	550 m	MAC protocol:	802.11 with `RTS/CTS`
Antenna type:	OmniAnt.	Propagation model:	TwoRayGround

We implemented Algorithm 1 as a protocol for setting up a wireless network in ns2. We refer to this implementation as the Maximum Lifetime Spanner (MLS) protocol. MLS consists of Algorithm 1 and the method of setting the transmission power levels of each node after running Algorithm 1, as described in Section 3. At start, each node is assumed to know link cost to each of its neighbours. This input can be obtained by the beaconing algorithm in Section 5. For simplicity, we use Euclidean distance as link cost in the simulation. As the result of the algorithm, each node has a list of neighbours for whose reachability it is responsible, which the node then uses to set its transmission power.

Both MLS and DMMT give trees that form optimal p-spanners by setting transmission power levels as described above, for each of the network instances. In our ns2 simulations, both algorithms converged despite a number of control packets being dropped by the MAC layer due to different reasons, such as network interference. Figure 2 shows the total (simulated) times required by the algorithms for convergence, where MLS is run on both the original disk graph (ODG) and on the RNG of the graph. As DMMT was insensitive to which input graph is used, for it only results on the original disk graph are given.

Our results indicate that MLS outperforms DMMT both in runtime and in the number of control messages transmitted, in particular when run on the RNG. MLS scales well with the number of nodes in the network, while DMMT shows a significant increase in the number of control messages required. However, the runtime of DMMT depends heavily on the values used to initialise the timers in the protocol, although the number of required control messages is unchanged.

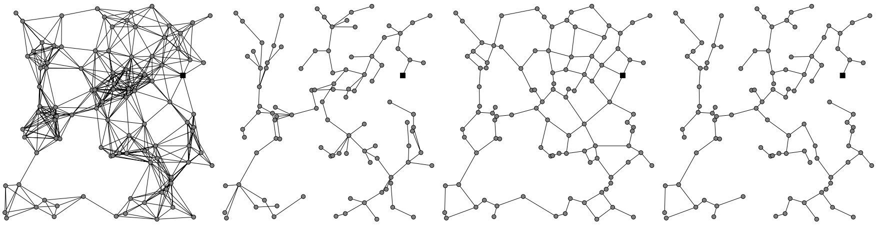
Running MLS on the RNG instead of the original graph reduces the number of messages required, as indicated by Fig. 2, but it also removes paths with low minmax cost and a small hopcount. Indeed, the experiments show a slightly higher running time, as propagating ACKs and NAKs along the tree takes longer.



(a) Total running time

(b) Total number of messages

Fig. 2. Simulated running time (s) and number of messages required by MLS and DMMT on networks of varying size. Errorbars represent the standard deviation; for MLS results are shown for runs on the original disk graph (ODG) and on the RNG. Note the logarithmic scale in (a).



(a) Initial transmission graph (b) Tree constructed by MLS on initial graph (c) RNG of original graph (d) Tree constructed by MLS on RNG

Fig. 3. Resulting minmax-power paths from the reference to the sensor nodes for a graph with 100 nodes; the remaining edges of the p-spanner induced by the corresponding transmission power are omitted for clarity

Figure 3 depicts one transmission graph instance, its RNG, and the tree that is constructed by MLS to calculate the transmission power levels for an optimal p-spanner for a network of 100 nodes.

7 Conclusions

We formulate the problem of lifetime maximisation in wireless sensor networks as a search for spanning subgraphs with minmax edge costs, which we call optimal p-spanners. We propose the MLS network protocol that determines the paths with minimum maximum edge cost. The algorithm is based on breadth-first search and is substantially simpler than methods relying on distributed minimum spanning tree algorithms. The `ns2` network simulator was used to compare the performance of MLS and DMMT in constructing minmax trees. In all scenarios considered MLS clearly outperforms DMMT in terms of number of control messages and execution time.

We also propose a distributed algorithm for extracting proximity graph structures. It uses beaconing to construct the RNG of the transmission graph of the network. We discuss the application of the algorithm to lifetime maximisation by integrating into a pre-processing stage before running the algorithm for finding optimal p-spanners. The resulting pruning of edges suggests significant gain in the efficiency of the original algorithm for dense networks.

To obtain meaningful results in practice, nodes must estimate the path loss for transmissions to their neighbours to a high accuracy. Furthermore, the path loss between neighbouring nodes has to be close to symmetric, as the edges in the spanning subgraph resulting from Algorithm 1 are likely to be used in the opposite direction than they were added during the construction of the optimal p-spanner. A pairwise exchange of link cost information would remove the need for this assumption.

In the future we will integrate the RNG construction by beaconing into the MLS protocol. We also hope to consider distributed approximation algorithms for dynamic transmission power assignment. One extension that readily lends itself to the problem of dynamic power assignment is an iterative method based on single scaled subproblems for the static case.

References

1. Ephremides, A.: Energy concerns in wireless networks. IEEE Wireless Comm. 9(4), 48–59 (2002)
2. Lloyd, E.L., Liu, R., Marathe, M.V., Ramanathan, R., Ravi, S.: Algorithmic aspects of topology control problems for ad hoc networks. Mobile Networks and Appl. 10(1-2), 19–34 (2005)
3. Ramanathan, R., Hain, R.: Topology control of multihop wireless networks using transmit power adjustment. In: Proc. 19th Annual Joint Conf. IEEE Comp. and Comm. Societies, pp. 404–413 (2000)
4. Kang, I., Poovendran, R.: Maximizing network lifetime of broadcasting over wireless stationary ad hoc networks. Mobile Networks and Appl. 10(6), 879–896 (2005)
5. Gallager, R.G., Humblet, P.A., Spira, P.M.: A distributed algorithm for minimum-weight spanning trees. ACM Trans. on Programming Languages and Systems 5(1), 66–77 (1983)
6. Rodoplu, V., Meng, T.H.: Minimum energy mobile wireless networks. IEEE J. on Selected Areas in Comm. 17(8), 1333–1344 (1999)
7. Wattenhofer, R., Li, L., Bahl, P., Wang, Y.M.: Distributed topology control for power efficient operation in multihop wireless ad hoc networks. In: Proc. 20th Annual Joint Conf. IEEE Comp. and Comm. Societies, pp. 1388–1397 (2001)
8. Gupta, S.K.S., Srimani, P.K.: Self-stabilizing multicast protocols for ad hoc networks. J. of Parallel and Distributed Computing 63(1), 87–96 (2003)
9. Guo, S., Yang, O.W.W., Leung, V.C.M.: Tree-based distributed multicast algorithms for directional communications and lifetime optimization in wireless ad hoc networks. EURASIP J. on Wireless Comm. and Networking 2007, 10 (2007) Article ID 98938
10. McCanne, S., Floyd, S., Fall, K., Varadhan, K.: The network simulator ns2 (1995) The VINT project, available for download at http://www.isi.edu/nsnam/ns/
11. Chang, J.H., Tassiulas, L.: Energy conserving routing in wireless ad-hoc networks. In: Proc. 19th Annual Joint Conf. IEEE Comp. and Comm. Societies, pp. 22–31 (2000)
12. Floréen, P., Kaski, P., Kohonen, J., Orponen, P.: Lifetime maximization for multicasting in energy-constrained wireless networks. IEEE J. on Selected Areas in Comm. 23(1), 117–126 (2005)
13. Lynch, N.A.: Distributed Algorithms. Morgan Kaufmann, USA (1996)
14. Toussaint, G.T.: The relative neighbourhood graph of a finite planar set. Pattern Recognition 12, 261–268 (1980)
15. Borbash, S., Jennings, E.: Distributed topology control algorithm for multihop wireless networks. In: Proc. 2002 Intl. Joint Conf. on Neural Networks (2002)
16. Bhardwaj, M., Misra, S., Xue, G.: Distributed topology control in wireless ad hoc networks using β-skeletons. In: Workshop on High Performance Switching and Routing, pp. 371–375 (2005)
17. Escalante, O., Pérez, T., Solano, J., Stojmenovic, I.: RNG-based searching and broadcasting algorithms over internet graphs and peer-to-peer computing systems. In: The 3rd ACS/IEEE Intl. Conf. on Computer Systems and Appl., pp. 47–54 (2005)

Maximizing Network Lifetime for Target Coverage Problem in Heterogeneous Wireless Sensor Networks[*]

Zheng Liu

School of Computer Science and Technology, Shandong Economic University,
Ji'nan, shandong, 250014, P.R. China
Lzh_48@126.com

Abstract. This paper presents an energy-efficient distributed target coverage algorithm(EDTC) for heterogeneous wireless sensor networks(HWSN) with multiple sensing units. In order to utilize the energy more efficiently, the sensor priority is introduced in this paper to integrate the sensing ability and the remaining energy together. EDTC is locally and simultaneously carried out at each sensor in a rounding fashion. Each sensor decides the on/off status of its sensing units at the beginning of each round, and then broadcasts the decision to its one-hop neighbors. The higher the priority of a sensor is, the shorter the decision time it needs. Simulation results show that compared with Energy First(EF) scheme and Integer Linear Programming(ILP) solution, EDTC has longer network lifetime than EF, and the performance difference between EDTC and ILP solution is confined within 10%.

1 Introduction

Wireless sensor networks(WSN) have recently attracted much attention of researchers due to its wide range of applications. A sensor is a tiny battery-equipped device capable of sensing, communication and computation. However, wireless sensor networks are power constrained. Therefore, it is desired to conserve energy so that the network lifetime can be maximized. Coverage problem is a fundamental issue in wireless sensor networks. It deals with problems of deploying wireless sensors and constructing sensor networks to cover regions of interest to detect. According to the kind of the regions that should be covered, the coverage problem can be classified as *Area Coverage Problem* and *Target Coverage Problem*. The area coverage problem aims at gathering information about an entire region. In contrast, the target coverage problem concerns about monitoring the state of a set of specific locations in the region.

There have been lots of efforts for target coverage problem for WSN, but most of the previous studies concentrated on homogeneous wireless sensor networks with single sensing unit based on centralized policies. Chinh T. proposed the energy-efficient scheduling for k-coverage problem based on single sensing unit[1].

[*] This work was supported by the National Natural Science Foundation of China (Grant No.60573181 and No.60673151).

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 457–468, 2007.

M. Cardei converted the target problem to maximal set cover problem and designed heuristic algorithms with centralized policies[5]. As presented in [6], M. Cardei introduced adjustable range set cover problem to extend network lifetime in the adjustable sensing ranges WSN. However M. Cardei's works did not consider the multiple sensing units. The approximation algorithm of K-coverage problem was discussed in [7][8] without multiple sensing units taken into account as well.

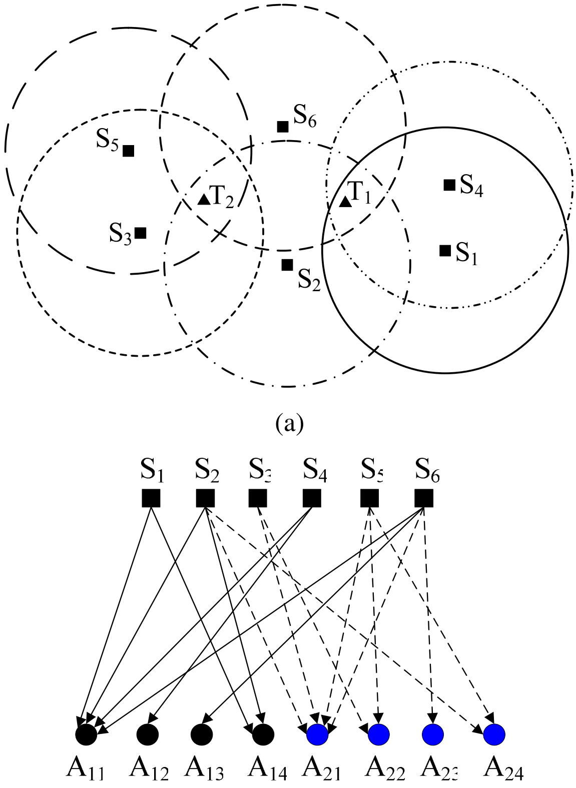
This paper addresses the energy-efficient target coverage problem in HWSN. A distributed target coverage algorithm for HWSN with multiple sensing units is presented to save the energy and prolong the network lifetime. The main principle behind EDTC is to introduce the concept of sensor priority, which is obtained by integrating two parameters together, which are the sensing ability and the remaining energy. For two sensors, if there is at least one parameter of a sensor different from the other one, the priority of the two sensors are unequal. In EDTC, the larger the value of the sensing ability is, or the larger the value of the remaining energy is, the higher the priority of the senor is. To the best of our knowledge, combining the two parameters to get a more reasonable policy has not been discussed.

The rest of the paper is organized as follows. Section 2 describes the target coverage problem and presents the ILP model. EDTC algorithm is presented in section 3. In section 4, an example is put forward to explain the algorithm in section 3. Section 5 presents the experimental results to demonstrate the performance of EDTC. Section 6 concludes the paper and points out the future work.

2 Problem Description and ILP Model

In order to simplify the target coverage problem, we make some restrictions and assumptions. Considering a number of targets with known locations that need to be continuously observed (covered) and a large number of sensors randomly deployed closed to the targets. We also assume the sensors have location determination capabilities (e.g. GPS), and all the sensors are built up randomly and can not be removed freely. Particularly, all the sensors have the same communication ability, computing ability and initial energy. Furthermore, supposing every sensor's communication distance is longer than twice of its sensing range so as to guarantee two sensors which can sense the same object can directly communicate to each other(that is, they are one-hop neighbors).

The wireless sensor networks discussed in this paper are heterogeneous and equipped with multiple sensing units, which means each sensor in the HWSN may be equipped with more than one sensing unit and the attribute each sensing unit can sense may be different as well. The number of sensors deployed in the sensing field is assumed to be larger than the optimum number needed to perform the target covering, so it is of great importance to decide correctly by which sensor the sensing attribute should be covered to make the power consumption minimized. An example(shown in Fig. 1(a)) is given with two targets and six sensors(with multiple sensing units) to illustrate the architecture of HWSN. The sensing area of a sensor is modeled as a circle centered at the sensor with radius as the sensing range. The bipartite graph shown in Fig.1(b) illustrates the corresponding coverage relationship in terms of sensing units on sensors and attributes at targets. The explanation of parameters in Fig.1 refers to section 3.2.



(a)

(b)

Fig. 1. An example with two targets and six sensors

Definition 1. *A target in an area A is said to be covered only if it is within the sensor's sensing range.*

Definition 2. *Energy-efficient Target Coverage Problem for Heterogeneous Wireless Sensor Networks: Given a two-dimensional area A and a set of N heterogeneous wireless sensors* $S=\{S_1, S_2, \ldots\ldots, S_N\}$*, derive an schedule from every sensor to determine the on/off status of its sensing units such that: 1) the whole area A is covered. 2) The network lifetime is maximized.*

Definition 3. *Network lifetime is the duration during which the whole monitored targets are covered.*

Target coverage in HWSN can be converted to Set Cover problem, then each set cover denotes the on/off status of sensing units in HWSN[5]. In [7], the coverage problem is formulated as integer linear programming(ILP) to obtain the optimal results. It is well known that solving ILP is an NP-complete problem. Due to limited energy and computing ability, it is impractical to use ILP to solve the HWSN target coverage problem on sensors. Therefore, we design a distributed heuristic algorithm for HWSN target coverage problem.

3 Energy-Efficient Distributed Target Coverage Algorithm

3.1 Algorithm Overview

The network activity is organized in rounds, meaning that each sensor runs this algorithm at the interval of a round time unit. Each round is made up of two phases, which are the initial phase and the sensing phase(shown in Fig. 2). In the initial phase, all the sensors have to decide which sensing unit should be turned on in the sensing phase. To guarantee all the sensors' activities can be synchronized, we assume all the sensors have a clock with a uniform starting time.

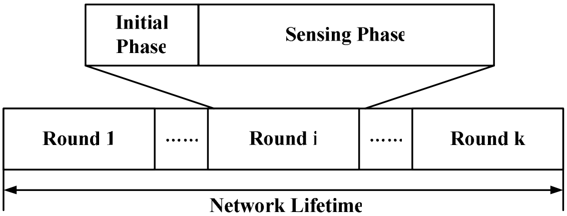


Fig. 2. Organization of network lifetime

3.2 Parameters Definition

To mathematically formulate this problem, the following notations need to be stated:

- M: The number of targets in sensing field.
- N: The number of sensors in sensing field.
- P: The number of sensing attributes.
- S_i:$(i=1,2......N)$: The i^{th} senor.
- $SA\ _i(i=1,2......N)$: The sensing ability of sensor i.
- $\alpha_k(k=1,2......P)$: The k^{th} sensing attribute.
- $T_j(i=1,2......\ M)$: The j^{th} target.
- A_{jk}: The sensing attribute α_k of target T_j.
- θ_i: The sensing attribute set in which the attributes can only be covered by S_i.
- θ_i^l : The l^{th} sensing attribute in θ_i.
- $E(\Delta)$: The energy consumption of the sensing attribute Δ in a round time.
- $E_i(i=1,2......N)$: The remaining energy of S_i.

- σ_i : The sensing attribute set in which the attributes can be covered by S_i excluding the ones in θ_i.
- $NS\left(\sigma_i^l\right)$: The number of sensors which can cover the l^{th} sensing attribute in σ_i .
- β_i : The sensing attribute set in which the attributes will be covered by S_i in the sensing phase of the current round.
- *DT*: The duration of initial phase.
- *ST*: The duration of sensing phase.
- DT_i: The decision time of S_i.
- P_i: The priority of S_i. The higher priority of a sensor, the shorter time for it to determine the status of its sensing units.
- DM_i: The decision message of S_i.
- NB_i: The sensor set in which the sensors is the one-hop neighbors of S_i.
- δ: The parameter which denotes the influence of the sensing ability on the sensor priority. The larger the value of δ is, the much more influence exerted on the sensor priority by the sensing ability is.

3.3 Computing Sensor's Priority

For the sake of prolonging the network lifetime, the sensor's energy should be used efficiently. Two parameters should be considered carefully, which are the remaining energy and the sensing ability of the sensor. In the former researches, the two parameters have been put forward in some papers[1][5][6]. However, combining them to obtain a more reasonable policy has not been discussed.

To compute the sensor's priority, we should obtain its sensing ability at first. The value of SA_i can be computed as followings.

$$SA_i = \sum_{m=1}^{|\sigma_i|} \frac{N}{NS\left(\sigma_i^m\right)} \quad . \tag{1}$$

As the attributes in θ_i can only be covered by S_i, the sensing units of S_i which can cover the attributes in θ_i must be turned on. Thus, there is no need to consider the attributes in θ_i.

In order to make the sensor priority decision more reasonable, we design a scheme to integrate SA_i and E_i together to compute the priority of S_i. There are two main principles in our scheme:

- The larger the value of SA_i is, or the larger the value of E_i is, the higher the priority of S_i is.
- For any two sensors, if there is at least one parameter of the sensing ability and the remaining energy unequal, the priority of them are different as well.

Based on the two principles illustrated above, Eq.(2) and Eq.(3) are designed to compute P_i.

$$\varepsilon_i = \left\lceil \frac{E_i - 2}{\delta} \right\rceil \quad . \tag{2}$$

$$P_i = \frac{(SA_i+\varepsilon_i)*\left[(SA_i-1-\varepsilon_i)*\delta+2E_i-2\right]}{2}+SA_i \ . \tag{3}$$

We consider two characteristics of the sensing ability and the remaining energy in a priority table which is computed in advance and saved in sensors localized. The ranges of these two parameters are divided into a number of different intervals, every which is represented by choosing a typical value. For any sensor with any sensing ability *SA* and remaining energy *E*, if *SA* and *E* is a typical relative sensing ability and a typical remaining energy respectively, the sensor's priority can be obtained by checking the priority table. If the two sensors' priority are same, the sensor with a smaller ID wins. Fig.3 shows the priority values when *δ*=2. In section 4 and 5, we use *δ*=2 to compute sensors' priority. The value of *δ* can be adjusted when needed.

SA									
5	25								
4	16	20	24						
3	9	12	15	19	23				
2	4	6	8	11	14	18	22		
1	1	2	3	5	7	10	13	17	
	1	2	3	4	5	6	7	8	E

Fig. 3. Priority table when δ=2

If one of *SA* and *E* is not among the typical values, the priority of the sensor can be calculated by interposing on the priority table. When the value of SA or E is arbitrary, we compute the sensor priority with the two-element three-point lagrange interpolation. To compute the priority of S_t whose sensing ability SA_t and remaining energy E_t are not among the typical values, we look for three typical values of the sensing ability which are the most adjacent ones to SA_t (we denote them SA_m(m=g, g+*1 and* g+*2*)). Similarly, we also seek three typical remaining energy values which are the nearest ones to E_t (we denote them E_n(n=h, h+*1 and* h+*2*)).

$$P_t=\sum_{m=g}^{g+2}\sum_{n=h}^{h+2}\left(\prod_{\substack{u=g\\u\neq m}}^{g+2}\frac{SA_t-SA_u}{SA_m-SA_u}\right)\left(\prod_{\substack{v=h\\v\neq n}}^{h+2}\frac{E_t-E_v}{E_n-E_v}\right)P(SA_m,E_n) \ . \tag{4}$$

3.4 Algorithm Description

This algorithm runs on each sensor in the initial phase of each round distributively to decide the on/off status of each sensing unit in the sensing phase. For S_i, the algorithm can be stated as follows.

Energy-efficient Distributed Target Coverage Algorithm (EDTC)

```
Input : { $E_i, \theta_i, \sigma_i, NB_i$ }
Output : Decide which sensing units of $S_i$ should be turned on and broadcast the decision message to its one-hop neighbors.
/*preparation phase*/
$SA_i = \sum_{m=1}^{|\sigma_i|} \frac{N}{NS\left(\sigma_i^m\right)}$;
/*For $S_i$, executing the following steps in the initial phase of each round*/
$E_i = E_i - \sum_{l=1}^{|\theta_i|} E\left(\theta_i^l\right)$;
computing $P_i$ and exchanging it with one-hop neighbors (please refer to section 3.3);
$DT_i = \left(1 - \frac{P_i}{max\left\{\left\{P_w \mid \forall w, S_w \in NB_i\right\} \cup P_i\right\}}\right) DT$;
$\lambda = NB_i$;
$\beta_i = \sigma_i$;
/*Listening to neighbors continuously for $DT_i$ time to make its own decision*/
while $DT_i$ is not expired do
    if receiving $DM_y$ then
        $\beta_i = \beta_i - \left(\beta_y \cap \beta_i\right)$;
        $\lambda = \lambda - S_y$;
    end if
end while
if $E_i < \sum_{l=1}^{|\sigma_i|} E\left(\sigma_i^l\right)$ then /* The remaining energy is not sufficient*/
    return coverage can't continue;
else
    if $\beta_i \neq \emptyset$ or $\theta_i \neq \emptyset$ then
    turn on the sensing units which can cover the attributes in $\beta_i$ and $\theta_i$;
    broadcast $DM_i$ to one-hop neighbors of $S_i$ which belong to the set $\lambda$;
    end if
end if
```

In order to reduce the communication complexity, there is no need to send *DM* to the neighbors which have been entered into the sensing phase. Let ω be the maximum number of one-hop neighbors that a sensor may have. The communication complexity can be estimated by the number of messages which are sent among all the sensors. For a sensor, the maximum messages it should send to its one-hop neighbors is no more than ω. So each sensor sends at most $O(\omega)$ messages in the initial phase. From the analysis above, we can draw the conclusions that the communication complexity of EDTC is $O(N\omega)$.

4 Example

In this section, we turn to a description of the target coverage algorithm illustrated above. The network topology of the example is described in Fig. 1(a) and the relationship between the sensors and the sensing units is shown in Fig. 1(b). We assume the initial energy of S_1-S_6 are same, the value of which is 24 units. The energy consumption of α_1-α_4 are 1, 2, 2, 3 respectively. We use the optimization toolbox in Matlab to solve this example through ILP solution. The optimal network lifetime is 12 rounds for this example. The first and second round are shown in Fig.4, which are repeated in the remainder rounds.

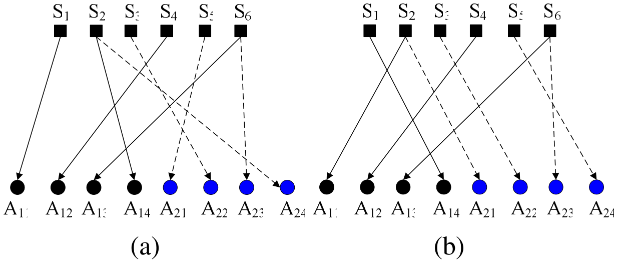


Fig. 4. The first and second round with ILP solution

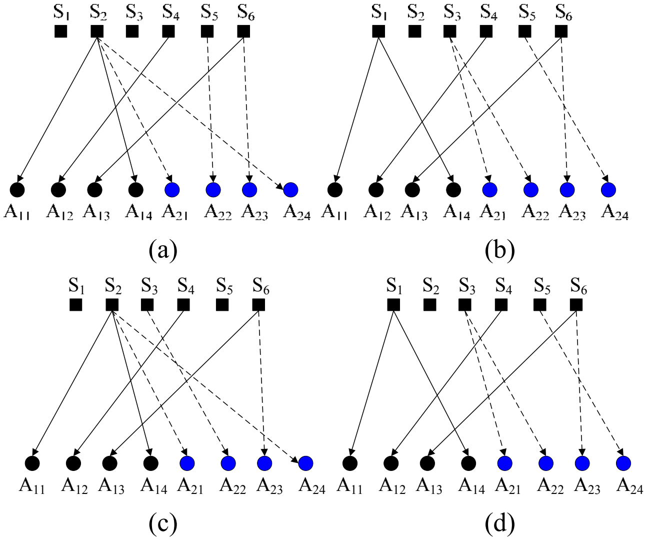


Fig. 5. The first four rounds with EDTC

In the following parts of this section, we solve this example using EDTC. Computing the sensing ability with Eq.(1), we obtain the value of SA_1- SA_6 are 4.5, 9, 4.5, 1.5, 7.5, 3, and the value of E_1- E_6 before the first round are 24, 24, 24, 22, 24, 22(excluding the energy consumption of the attributes in θ_i). Consequently, the priority ranking is $P_2>P_5>P_1>P_3>P_6>P_4$ computed by Eq.(2)(3)(4). Therefore, we obtain the on/off status of each sensing unit in the first round(shown in Fig.5(a)). By the same reason, we have the priority ranking in the second round which is $P_1>P_3>P_5>P_2>P_6>P_4$. The on/off status of each sensing unit is shown in Fig.5(b). Similarly, the third and fourth round are shown in Fig.5(c)(d). The first four rounds are repeated in the remainder rounds. Finally, network lifetime computed by EDTC is 12 rounds which is equal to the optimal results.

5 Experimental Results

We develop a simulator with C++ to evaluate the efficiency of EDTC through conducting some simulations measuring the network lifetime with different number of sensors and different number of targets and also different number of sensing attributes. Our simulations are based on a stationary network with sensor nodes and targets randomly located in a 400m×400m area. We assume the sensing units are randomly assigned to sensors with the number not exceeding upper bound which is set in advance. Without loss of universality, the energy consumption of a sensing unit in a round is assumed equal to its ID. All the sensors are supposed to have the same initial energy(50 units). The sensing range of each sensing unit is set as 50m. The communication range of each sensor is twice of the sensing range. The initial phase lasts 8 seconds, and the duration of a round is 10 minutes.

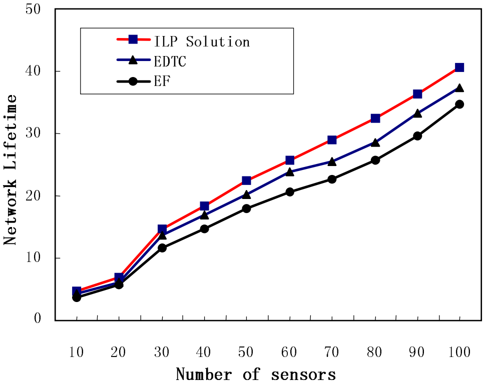


Fig. 6. Network lifetime with number of sensors

In the experiments, all the measurements are averaged over 15 runs to make the results more accurate. Energy consumption in communication and computation is omitted. In addition, a reliable communication channel is also assumed. The performance of EDTC is compared with ILP solution and energy first scheme(EF)[1] which is a greedy approach to make decisions for a sensor to enable its sensing units only considering its remaining energy. The ILP solution is implemented by the optimization toolbox in Matlab. The unit of network lifetime in our experiment is a round.

In experiment 1, we measure the network lifetime when the number of sensors varies between 10 and 100, and the number of targets and attributes are respectively fixed to 25 and 4. The network lifetime increases with the number of sensors, as more sensors provide more opportunities to cover the targets(shown in Fig. 6).

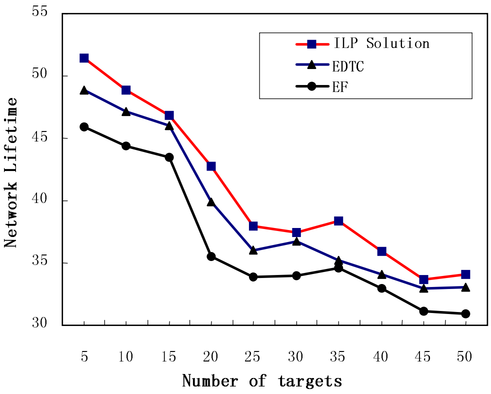


Fig. 7. Network lifetime with number of targets

Experiment 2 measures the impacts of the number of targets on the network lifetime, when the number of targets and attributes are respectively equal to 100 and 4. From the results we know the network lifetime decreases with the number of targets, because more targets consume more energy to cover them(shown in Fig. 7).

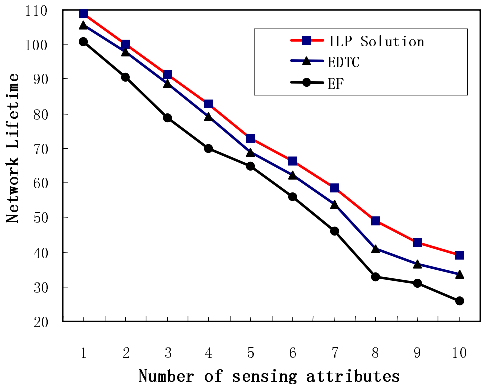


Fig. 8. Network lifetime with number of sensing attributes

Experiment 3 illustrates the relationship between network lifetime and the number of sensing attributes. We assume the number of sensors and targets are 200 and 25 respectively (shown in Fig. 8).

The simulation results can be summarized in Table 1. AT means the average network lifetime under a specific scheme. P denotes the proportion between a specific scheme's AT value and the AT value of ILP solution.

As the data shown in Table 1, EDTC effectively prolong the network lifetime compared with EF, and the difference of network lifetime between EDTC and ILP solution is no more than 10%.

Table 1. Experiment data

Scheme	Experiment 1		Experiment 2		Experiment 3	
	AT	P	AT	P	AT	P
ILP Solution	23.16	100%	40.75	100%	71.18	100%
EDTC	20.95	90.45%	39.01	95.72%	66.74	93.75%
EF	18.69	80.69%	36.66	89.97%	59.67	83.83%

6 Conclusions and Future Directions

This paper investigates the energy-efficient target coverage problem in HWSN with multiple sensing units. Distinct from the past work, this paper proposes a completely heuristic and distributed algorithm named EDTC to solve the target coverage problem. Simulation results are presented to verify the advantages of EDTC in the field of saving energy and prolonging network lifetime compared with ILP solution which is the optimal scheme and the existing method EF.

We have only taken a first step towards the energy-efficient target coverage problem in HWSN. For future research, we will consider that the sensing range can be adjusted freely, and the cases that the network topology changes dynamically, and calculate more factors which impact the energy consumption, so as to make the simulation results more accurate.

References

1. Vu, C.T., Gao, S.: Distributed Energy-Efficient Scheduling Approach for K-Coverage in Wireless Sensor Networks. In: Military Communications Conference, pp. 1–7 (2006)
2. Huang, C.-F., Tseng, Y.-C.: The Coverage Problem in a Wireless Sensor Network. In: Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications, San Diego, CA, USA, pp. 115–121 (2003)
3. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless Sensor Networks: a Survey. Computer Networks 38(4), 393–422 (2002)
4. Lee, J.-J., Krishnamachari, B., Kuo, C.J.: Impact of Heterogeneous Deployment on Lifetime Sensing Coverage in Sensor Networks. In: SECON. Proceedings of the IEEE International Conference on Sensor and Ad Hoc Communications and Networks, pp. 367–376 (2004)

5. Cardei, M., Thai, M.T.: Energy-efficient Target Coverage in Wireless Sensor Networks. In: INFOCOM. Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 3, pp. 1976–1984 (2005)
6. Cardei, M., Wu, J., Lu, M., Pervaiz, M.O.: Maximum Network Lifetime in Wireless Sensor Networks with Adjustable Sensing Ranges. In: WiMob. Proceedings of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, vol. 3, pp. 438–445 (2005)
7. Yang, S., Dai, F., Cardei, M., Wu, J.: On Multiple Point Coverage in Wireless Sensor Networks. In: MASS. Proceedings of the IEEE International Conference on Mobile Ad-hoc and Sensor Systems, pp. 757–764 (2005)
8. Wang, J., Zhong, N.: Efficient Point Coverage in Wireless Sensor Networks. Journal of Combinatorial Optimization 11(3), 291–304 (2006)
9. Lu, M., Wu, J., Cardei, M., Li, M.: Energy-efficient Connected Coverage of Discrete Targets in Wireless Sensor Networks. In: Lu, X., Zhao, W. (eds.) ICCNMC 2005. LNCS, vol. 3619, pp. 43–52. Springer, Heidelberg (2005)
10. Chang, J.-H., Tassiulas, L.: Maximum Lifetime Routing in Wireless Sensor Networks. IEEE/ACM Trans. on Networking 12(4), 609–619 (2004)
11. Younis, O., Fahmy, S.: Distributed Clustering in Ad-hoc Sensor Networks: a Hybrid, Energy-efficient Approach. In: Proceedings of INFOCOM 2004 (2004)
12. Mhatre, V.P., Rosenberg, C., Kofman, D., Mazumdar, R., Shroff, N.: A Minimum Cost Heterogeneous Sensor Network With a Lifetime Constraint. IEEE Trans. On Mobile Computing 4(1), 4–15 (2005)

Balancing Security and Energy Consumption in Wireless Sensor Networks

Chih-Chun Chang, David J. Nagel, and Sead Muftic

The George Washington University,
2033 K St. NW, Suite 340, Washington D.C., U.S.A
ccc987@hotmail.com, nagel@gwu.edu, sead@dsv.su.se

Abstract. Appling security to messages traveling over wireless links in sensor nodes requires additional energy. This paper describes our suggestions on balancing the level of security and energy consumption based on our measurements using CrossBow and Ember sensor nodes. It was found that the node microcontroller's CPU operates for substantially longer times for both hashing and encryption operations compared to the time for handling messages without any security. However, this has little overall impact on energy consumption. The longer high-power radio transmission times due to hashing were especially costly. For the full operational mode, with CPU processing and also radio transmission of messages, our results indicate that the lifetime of a transmitting node in a security regime is only about one-half of the lifetime without security. Hence, we provided design guidelines to apply security with energy consideration for WSN. They include 2 to 8 bytes MACs for integrity and authentication instead of SHA-1, and the size of messages should match the steps of encryption algorithms.

Keywords: Wireless sensor networks (WSN), energy cost, security, experimentation, and measurement.

1 Introduction

Wireless sensor networks (WSN) are a rapidly emerging technology with great potentials for many different applications. Such networks are a result of relentless integration of various technologies: sensors (usually specific to an application), microcontrollers, radio transceivers and other components [1,2]. A defining character of wireless sensor networks is their resource limitations. In general, they have few excess capabilities in order to conserve the energy available from their batteries and, accordingly, extend their life times. WSNs are vulnerable to attacks, which are more difficult to launch in a wired network, since they use wireless communication. Security services such as integrity, confidentiality and node authenticity are critical for preventing intruders, adversary nodes or anybody else from compromising actions of a distributed sensor network. But, security in wireless sensor networks is still a very young field with many challenges and opportunities.

Most commercial WSNs do not provide any security for their messages, because it introduces complexity and requires additional energy. Hence, the question arises:

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 469–480, 2007.

what is the cost of energy required when adding security to wireless sensor networks? That is, by how much are the battery and network lifetimes shortened when using protection of the transmitted sensor data messages? Alternatively, how to select and apply efficiently security algorithms for WSN in terms of reducing additional energy consumption? The primary goal of this chapter is to describe solutions to these questions.

We start with determining the energy consumption required for execution of various security algorithms and for transmitting longer messages on the top of the baseline, indicated in Fig. 1. By “Level of Security” we mean what algorithms are used, such as hash–only, encryption–only, or hash with encryption. Also, the strength of cryptographic algorithms usually can be determined by their number of keys, key sizes or the number of internal iterations. The baseline for comparison is consumption of energy for operations without security. As the level of security is increased by the use of more robust algorithms, the times required for operations of the controller and for radio transmission (or reception) of longer messages both increase. The variation of energy consumption (required power) with security is shown Fig. 1 as continuous and linear. However, because security algorithms vary discontinuously in their type and characteristics, the relationship is not that simple. But, increasing security does require increased consumption of energy. We investigated the relationship for particular combinations of nodes and algorithms.

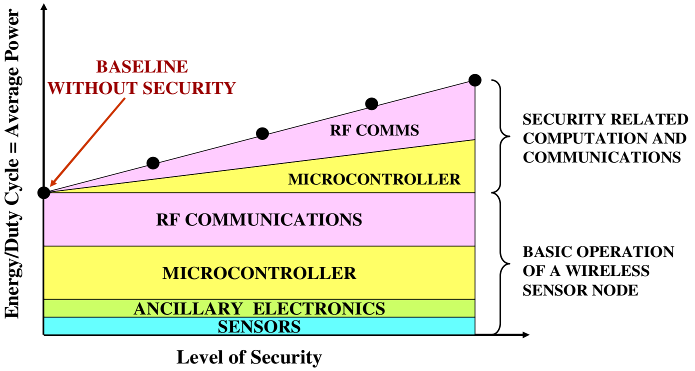


Fig 1. Schematic representation of the four components of sensor nodes that draw power from the battery. Increased energy consumption for security, due to additional CPU & radio operation, is indicated schematically. By “level of security” we mean what algorithms are used and their characteristics.

The more capable controllers and transceivers in sensor nodes permit the preparation and transmission of messages with greater security. Of course, energy consumption and node capabilities are also related, so a three-dimensional presentation of the relationships between all these factors can be made. This is indicated in Fig. 2. Therefore, the level of security is a trade-off between increased

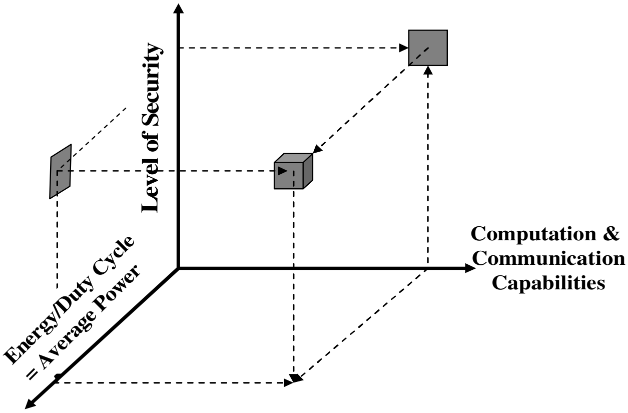

Level of Security
Computation & Communication Capabilities
Energy/Duty Cycle = Average Power

Fig. 2. Schematic relationship of the levels of security possible in the nodes of wireless sensor networks with the associated energy costs and required node capabilities

consumption of energy, due to extended computation and transmission times, and node characteristics, especially the size of available memory.

2 Wireless Sensor Networks, Crypto Algorithms and Measurement Techniques Employed

Two commercial wireless sensor networks were used in our work, CrossBow MICA2 and Ember EM2420. They have the same ATmega128L microprocessor with 128K bytes code memory and 4K bytes data memory. The software from the two companies was very different, with CrossBow using the Tiny OS and their own communications protocol, and Ember not having an operating system and using ZigBee protocol. They were different in programming language and compilers, too.

We found that prevailing limiting factor for security extensions was the data memory. Since nodes with core operations, networking, sensor and sink functions need to share 4K bytes data memory, the room left for security extensions was very limited. Hence, we developed an innovative approach for usage of data memory in sensor nodes. The essence of the method is to re-use the same portions of memory during execution of a cryptographic algorithm. For that, the code of each full strength algorithm must be carefully restructured into separate and autonomous smaller sections.

Using this method, we reorganized and successfully ported well-known, standard security algorithms into sensor nodes: SHA-1, RC-5, DES, and AES. It is important to emphasize that the functionality and results of these algorithms were not modified or reduced. After employing the new technique, we were able to execute hashing and encryption algorithms in the CrossBow and Ember nodes as shown in Fig. 3. The details of this method and memory usage for these operations are given in another paper [3].

Platform Algorithm	**CrossBow MICA2** and the GCC Compiler	**Ember EM2420** and the IAR Compiler
Hashing	**SHA-1:** OK	**SHA-1:** OK
Encryption	**RC5:** OK **DES-CBC:** X2, 4X or X8 OK (Divide and Conquer technique) **AES:** OK (Divide and Conquer technique)	**RC5:** OK **DES-CBC:** X2, 4X or X8 OK (Divide and Conquer technique) **AES: Not OK** IAR compiler used too much ROM

Fig. 3. Hashing and encryption algorithms executed in the indicated nodes

In order to find out the additional energy needed for these security algorithms, we measured the energy consumed without and with security, where the difference represents the extra energy required for security. We used the circuit diagram showed in Fig. 4 and a PicoScope 3206 [4] for the power and time measurements.

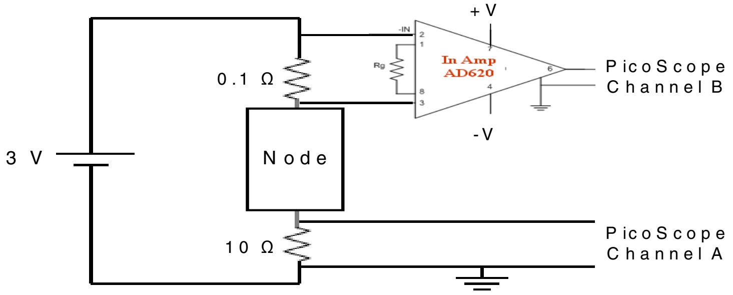


Fig. 4. Measurement circuit with 0.1 Ω and 10 Ω sense resistors both without (*channel A*) and with amplification using the AD620 instrumentation amplifier (*channel B*)

Two approaches, without amplifier and with amplifier, were used in order to help insure that the voltage measurements are correct. An instrumentation amplifier was chosen because it is sensitive only to differences in the two inputs, and eliminates noise common to both inputs.

Then, the energies (E) were calculated from the powers (P), required to run the micro-controller and radio for specific periods of time (T), using the formula E = P x T. The power and time were obtained by measuring the currents (I) from the batteries to the controller and radio, using a sense resistor R in the circuit. The currents were computed from the measured voltages (V) across the sense resistor from Ohm's Law, namely I = V/R. Then, P = I x BV, where BV is the voltage of the batteries.

3 Energy Costs for Security

As noted earlier, the energy consumed by the sensors and their ancillary electronics is not dependent on the use of security. However, the energies consumed by the controller and transceiver are both sensitive to execution of security algorithms and their results.

We first measured execution times for processes by the controller and for radio transmission without security, since those results were used as the baseline for measurements when using security algorithms. The details of measurement for all operations are given in another paper [5]. We found that the differences between CrossBow and Ember transmission characteristics go well beyond the baud rate. A 29-byte message in a CrossBow network requires 28.5 msec for transmission, compared to 2.5 msec for the Ember technology. Since the radio transmission powers in the two networks are comparable, the conclusion is that RF transmission consumes much less energy in an Ember network than in a CrossBow network.

We loaded and measured execution times for the hashing algorithm SHA-1, and then the encryption algorithms RC5, DES-CBC and AES 128, all as a function of message length. The SHA-1 algorithm adds 20 bytes to the message length, so that the energy consumption is greater since the radio transmit extra bytes. It must also be noted that a message data payload of 29 bytes is the maximum for a single transmission in CrossBow nodes and 68 bytes for Ember nodes. Hence, longer messages can require multiple packet transmissions. This imposes a significant energy penalty. Adding hashing to the unencrypted or encrypted messages has the same impact, since encryption we used does not increase the length of the original message.

It is also found the hashing algorithms SHA-1, encryption algorithms RC5, DES-CBC and AES- 128 all have similar behavior with increasing message length. SHA-1 had a constant step size of every 64 bytes. The step increases occur in RC5 and DES-CBC at message lengths of 8N (N = 1, 2, 3….) bytes in both nodes. In the case of AES-128, the execution times were constant to a message length of 128 bytes.

3.1 Energy Cost for Security in CrossBow Nodes

The measured energy costs in CrossBow nodes without and with security are summarized on an absolute basis in Table 1 and on a relative basis in Table 2. Examination of Table 1 produces the following conclusions. For an 8 byte message,

Table 1. Experimental absolute energy costs in μJ for four message lengths on CrossBow nodes

Message Length (Bytes)			8	16	24	32
No Security	CPU		3	4	4	4
	Transmit		945	1113	1281	2226
CPU and Transmit			948	1117	1285	2230
Hash	CPU	SHA-1	154	154	154	154
	Transmit		2142	2310	2478	3423
Hash and Transmit			2296	2464	2632	3577
Encrypt	CPU	RC5	111	124	137	150
	CPU	DES	53	79	103	126
	CPU	AES128	339	339	339	339
Encrypt and Transmit		RC5	1056	1237	1418	2376
		DES	998	1192	1384	2352
		AES128	1248	1452	1620	2565
Hash, Encrypt & Transmit		RC5	2253	2434	2615	3573
		DES	2195	2389	2581	3549
		AES128	2481	2649	2817	3762

the CPU operating without security requires only about 4 μJ, but 154 μJ are needed for hashing alone. For encryption without hashing, the required energy varies from 111 to 150 μJ for RC5, from 53 to 126 μJ for DES-CBC, and is 339 μJ for AES-128. That is, the CPU operates for substantially longer times for both hashing and encryption relative to the time required for message handling without any security. But, these seemingly dramatic increases are not so important, because the associated energies for CPU operations are not so large compared with the energies required for radio transmission.

Table 2. Experimental energies relative to the operation without security on CrossBow nodes

Message Length (Bytes)			**8**	**16**	**24**	**32**
No Security: CPU and Transmit			1.00	1.00	1.00	1.00
Hash and Transmit			2.42	2.21	2.05	1.60
Encrypt	CPU	RC5	0.12	0.11	0.11	0.07
	CPU	DESCBC	0.06	0.07	0.08	0.06
	CPU	AES 128	0.36	0.36	0.36	0.36
Encrypt and Transmit		RC5	1.11	1.11	1.10	1.06
		DESCBC	1.05	1.07	1.08	1.05
		AES 128	1.32	1.30	1.26	1.15
Hash, Encrypt & Transmit		RC5	2.38	2.18	2.03	1.60
		DESCBC	2.32	2.14	2.10	1.59
		AES 128	2.62	2.37	2.19	1.69

Without security, the radio transmission energies range from 945 to 2230 μJ for data messages with payloads of 8 to 32 bytes. If messages are hashed, the corresponding values are significantly higher, namely from 2296 to 3577 μJ. It can be noticed from Table 2 that the addition of hashing increases energy consumption for messages in the 8-32 byte range by factors of 1.60 to 2.42. The relative increases due to encryption and transmission without hashing are about 1.1 for the RC5, 1.1 for the DES-CBC and 1.2 to 1.3 for the AES algorithm. Hence, hashing and transmission requires more than twice the energy required for encryption and transmission, with the no-security case being the baseline.

If hashing, encryption and transmission are employed, the increases in energy consumption relative to the no-security case, are 1.6 to 2.4 for the RC5, 1.6 to 2.3 for the DES-CBC and 1.7 to 2.6 for the AES-128 algorithm. These numbers are the "bottom line" for the energy costs of adding security to the CrossBow wireless sensor network. That is, hashing, encryption and transmission on the sending nodes roughly halves the lifetime of the network batteries.

3.2 Energy Cost for Security in Ember Nodes

As was the case for the CrossBow nodes, thorough measurements were performed for the security algorithms that could be loaded into and run within the Ember nodes. The overall energy costs without and with security are summarized on an absolute basis in Table 3 and on a relative basis in Table 4.

Examination of Table 3 can produce the following conclusions. CPU operations without security require only about 6 µJ, but 75 µJ are needed only for hashing. For encryption without hashing, the energy values vary from 100 to 146 µJ for the RC5, and from 104 to 204 µJ for the DES-CBC algorithms. That is, the CPU operates for substantially longer times for both hashing and encryption relative to the time required for message handling without any security. As with CrossBow, these apparently large increases are not very important, because the energies for CPU operations are small compared with the energies for radio transmission.

Without security, the radio transmission energies range from 80 to 119 µJ for data messages with payloads of 8 to 32 bytes. These values are on the order of one-tenth of the transmission energies required by CrossBow, due to the higher baud rate of the Ember nodes. If messages are hashed, the corresponding values are significantly higher, namely 107 to 141 µJ. It can be seen from Table 4 that the addition of hashing increases energy consumption for messages in the 8-32 byte range by factors of 1.81 to 2.14. The relative increases due to encryption and transmission without hashing are about 1.2 for RC5 and 1.5 for DES-CBC. Hence, hashing takes about twice the energy as encryption, with the no-security case being the baseline.

Table 3. Experimental absolute energy costs in µJ for four message lengths on Ember nodes

Message Length (Bytes)			**8**	**16**	**24**	**32**
No Security	CPU		5	6	6	6
	Transmit		80	91	103	119
CPU and Transmit			85	97	109	119
Hash	CPU	SHA-1	75	75	75	75
	Transmit		107	120	128	141
Hash and Transmit			182	195	203	216
Encrypt	CPU	RC5	100	116	129	146
	CPU	DESCBC	104	138	171	204
Encrypt and Transmit		RC5	180	207	232	259
		DESCBC	184	229	274	317
Hash, Encrypt & Transmit		RC5	207	236	257	287
		DESCBC	211	258	299	345

Table 4. Experimental energies relative to the operation without security on Ember nodes

Message Length (Bytes)			8	16	24	32
No Security: CPU and Transmit			1.00	1.00	1.00	1.00
Hash and Transmit			2.14	2.01	1.86	1.81
Encrypt	CPU	RC5	1.18	1.20	1.18	1.23
	CPU	DESCBC	1.22	1.42	1.57	1.71
Encrypt and Transmit		RC5	2.12	2.13	2.13	2.18
		DESCBC	2.16	2.36	2.51	2.66
Hash, Encrypt & Transmit		RC5	2.43	2.42	2.36	2.41
		DESCBC	2.48	2.66	2.74	2.90

If hashing, encryption and transmission are employed, the increases in energy consumption relative to the no-security case are about 2.4 for RC5 and 2.48 to 2.90 for DES-CBC. These numbers are the "bottom line" for the energy costs of adding security to the Ember wireless sensor network. That is, hashing, encryption and transmission on the sending nodes roughly halves the lifetime of the batteries.

3.3 Comparisons of CrossBow and Ember Nodes

In this section, we compare the energy costs for adding security to both nodes. When the energies for transmissions are added to the energies required for the CPU operations for both nodes, the situation for the secure regime is similar to their individual operations without security. The difference in the transmission rates for nodes from the two companies translates into much more energy being required by CrossBow than by Ember. One can compare the energies for CrossBow and Ember for various combinations of hashing, encryption and transmission. However, we choose to compare the required energies for all of these functions, since they are commonly used together. The results are presented in Table 5.

Table 5. The absolute energies in µJ required for the combination of hashing, encryption (RC5 or DES-CBC) and transmission (Xmt) for CrossBow and Ember nodes

Message Length (Bytes)		8	16	24	32
SHA-1, RC5 & Xmt	CrossBow	2253	2434	2615	3573
	Ember	207	236	257	287
SHA-1, DES-CBC & Xmt	CrossBow	2195	2389	2581	3549
	Ember	211	258	299	345

Comparison of the energies for operation of the CPU and radio without any security shows that the CPU energies are comparable for both technologies. The energy required for the radio transmission is much less, roughly 5-8 %, for Ember compared to CrossBow nodes. This is clearly due to their similar (maximum) radio powers, with both near 50mW, but the very different transmission rates, namely 19.2 kbps for CrossBow and 250 kbps for Ember. This ratio of about 250/20 = 12.5 is generally consistent with the difference in the energy requirements of nodes from the two companies. In general, the slower radio rate in the CrossBow nodes translates into a factor of around 9 to 11 in the energy required for the three operations. For the 32 byte message length, which is beyond the payload size for CrossBow but not for Ember, the ratio is over 12 for the RC5 algorithm.

4 Guidelines to Apply Security into WSN

In our work, we have also tried to port many cryptographic algorithms and security services into several different platforms besides CrossBow MICA2 and Ember EM2420 nodes discussed above. These platforms include WSN nodes: i-Bean [6] from Millennial Net and M1010 [7] from Dust, and WSN gateway StarGate [8] from CrossBow.

The Dust wireless sensor nodes and the Millennial Net wireless sensor nodes do not allow any modifications at the link layer or message payload. Therefore, we could not port cryptographic algorithms into their devices.

StarGate is powerful single board computer and therefore can be used as WSN gateway. Although we could load and execute some cryptographic algorithms within the nodes from CrossBow and Ember, it was possible to do so for only a very limited set of the entire array of industrial strength hashing and encryption algorithms, which we used. This limited ability, and the inability to measure program execution times within the micro-controllers in the commercial sensor nodes, forced us to consider using a more capable computer, the StarGate. With Stargate, we ported and tested complete crypto engine based on openSSL. Therefore, we were able to generate the table of execution times for all algorithms and various message lengths.

The level of security for a WSN platform was obviously limited by its software, such as black-box, and hardware such as memory. Fig. 5 shows the results of available security services for different WSN platforms. We found that the RSA-1024 operation in StarGate took 5-20 seconds, which was not acceptable. Therefore, we put authentication "light", which means authentication process relies on hash and symmetric key encryption algorithms only, not based on RSA algorithms.

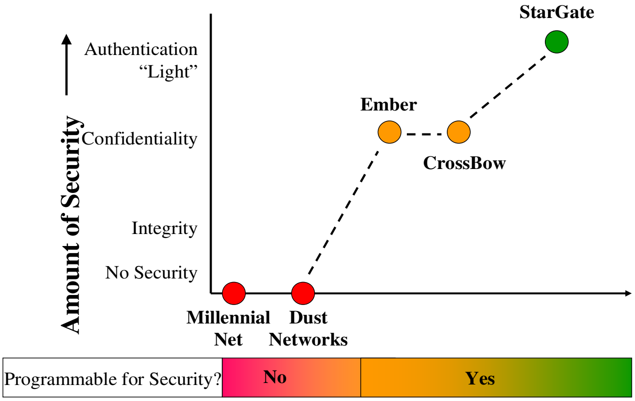


Fig. 5. The results of available security services for different WSN platforms

Based on the previous measurements results, it is easy to see that there are major tradeoffs between factors such as computing on a node and communicating between nodes, energy consumption, and the level of security.

In our research we discovered that the major factor causing significant energy consumption was transmitting extra message bytes. Since a low-power operational mode is always desired in WSN, we provide some design guidelines to apply security into the sensor network.

There are several other operational factors to be considered when applying security. Existing protocols, such as IPSec, SSL, and SSH, are too heavy-weight for use in sensor networks. Their packet formats add many bytes of overhead, and they were not designed to run on computationally constrained devices. The most basic factor is the fraction of the time that security is to be used. Then, the method how to apply security must also be considered. In some cases, hashing to insure message integrity may be all that is needed. In others, encryption may be sufficient. Other situations might require authentication, key management or other, more sophisticated security services, such as access control. Of course, various combinations of these types of security functions are possible with powerful WSN nodes.

The next consideration is the strength of algorithms applicable to each of the security services. Algorithms for hashing and encryption provide various levels of strengths of integrity and secrecy. Stronger algorithms require more computations and transmissions, so they consume more energy. In some scenarios, low-power operation without security might be possible until some event occurs. Then, the network might be programmed to switch into the higher-power mode with another type and level of security.

Hashing schemes. SHA and SHA-1 algorithms are defined in FIPS 180 and FIPS 180-1 standards. MD2, MD4, and MD5 are message-digest algorithms developed by Rivest. MD2, MD4, and MD5 algorithms take a message of arbitrary length and produce a 16-byte message digest, while SHA and SHA-1 take a message of less than 264 bits in length and produces a 20-byte message digest. This means, when hashing is applied to the message, extra 16 bytes or 20 bytes are appended and must be transmitted. If hash is applied to every sensor data packet with a size approximately 20 bytes, the result in energy cost will be double. Therefore, we suggest applying hash algorithms to a group of packets, not to individual packets.

To achieve message integrity and authentication in WSN we suggest using Message Authentication Codes (MACs). Since block encryption algorithms may already be available in the node, we can utilize those algorithms in CBC mode to create a message authentication code. The MAC algorithms can be designed to create MACs of 2 to 8 bytes for authentication of each packet, compared to 16 or 20 bytes for standard hashing algorithms.

Encryption schemes. Most of the encryption algorithms are implemented as block ciphers. A block cipher is a keyed pseudorandom permutation over small bit strings, typically 8 or 16 bytes. Examples of block ciphers are Skipjack, RC5, DES, and AES. For instance, DES has a constant step size of every 8 bytes. RC5 also steps up each 8 bytes, while AES has steps at 128 byte intervals. Therefore, it is good to match the WSN messages to constant step size to reduce extra energy consumption. It is also worthwhile to mention that these encryption algorithms do not generate extra bytes. If messages are longer than 8 or 16 bytes, block ciphers can be used in CBC mode to encrypt such messages.

5 Related Work

The overhead for energy consumption for security algorithms has been studied for general-purpose computing, such as the cost of SSL on PCs [9] or WEP for Wi-Fi [10]. As far as energy consumption for WSN is concerned, only a few studies have been conducted. Most studies of energy consumption for WSN are restricted to simulations. Coarse approximations of energy consumption are usually derived from the number of transmitted packets and CPU duty cycles. The network simulation tools, such as ns2, TOSSIM [11] and Atemu [12], are effective for understanding the behaviors of network protocols. However, they cannot capture the behavior of individual nodes in WSN in detail. A few instruction-level models to evaluate energy consumption for sensor network nodes have been developed, such as PowerTOSSIM [13] and AEON [14]. These two models are based on the measurement of node current, and then breakdown to individual source code routines and components to get final energy consumption. However, such approaches are not suitable for black-box software of most commercial WSN nodes. Also, it is a very tedious job to insert instruction-counting statements into all the blocks of source code.

Arvinderpal et al. [15] presented the energy cost of RSA and ECC algorithms on the MICA2DOT nodes. However, they did not load the full version of the RSA and ECC algorithms. The results in Gupta's paper [16] showed that the total memory required in RAM is very close to 4K bytes. This means there is no room left for sensing or networking applications. On the other hand, as we emphasized before, WSN messages are usually very short, approximately 20 bytes, so, if we apply the RSA-1024 algorithm, the total length of transmission data is 1024 bytes plus the 1024 bytes key, which is too large an overhead.

6 Conclusions

We acquired and operated two state-of-the-art commercial wireless sensor networks. We designed and validated a system to measure instantaneous power consumption for CPU operation and radio transmission. Experimental measurements of the electrical currents from the batteries to the CrossBow and Ember nodes permitted determination of the powers required for various functions, such as CPU operation and radio transmission. The duration of each such operation gave the times that were needed to compute the energy required for operations without or with security.

We demonstrated the ability of the measurement system to give power consumption results that compared favorably with values in the literature. Then, it was used to obtain absolute energies for operation of the security algorithms on data messages of different lengths, beyond the energies needed for computations and transmissions on messages with no security. Finally, we provided the guidelines to apply security into energy-constrained low-power WSN.

References

1. Akyildiz, I., Su, W., Sankarasubramaniam, Y., Cayirci, E.: A Survey on Sensor Networks. IEEE Communications Magazine, 102–114 (2002)
2. Chong, C., Kumar, S.: Sensor networks: Evolution, opportunities, and challenges. IEEE, Los Alamitos (2003)
3. Chang, C., Muftic, S., Nagel, D.J.: Cryptographic Algorithms in Wireless Sensor Nodes, submitted to Ad hoc & sensor wireless network journal (2007)
4. http://www.picotech.com/picoscope-3000.html
5. Chang, C., Nagel, D.J., Muftic, S.: Measurement of Energy Costs of Security in Wireless Sensor Nodes. In: The 16th IEEE ICCCN (2007)
6. http://www.millennial.net/products/meshscape916.asp

7. http://www.dust-inc.com/products/eval_kit.shtml
8. http://www.xbow.com/Products/productsdetails.aspx?sid=85
9. Badia, L.: Real world SSL benchmarking, Rainbow Technologies, Inc
10. Prasithsangaree, P., Krishnamurthy, P.: Analysis of energy consumption of RC4 and AES algorithms in wireless LANs. In: Global Telecommunications Conference, vol. 3, pp. 1445–1449 (2003)
11. Tossim, http://www.cs.berkeley.edu/~pal/research/tossim.html
12. Atemu, http://www.hynet.umd.edu/research/atemu/
13. Shnayder, V., Hempstead, M., Chen, B., Allen, G., Welsh, M.: Simulating the power consumption of large-scale sensor network applications. In: The 2nd international conference on Embedded networked sensor systems, Baltimore (2004)
14. Landsiedel, O., Wehrle, K.: Aeon: Accurate Prediction of Power Consumption in Sensor Networks. In: The Second IEEE Workshop on Embedded Networked Sensors, Sydney (2005)
15. Wander, A., Gura, N., Eberle, H., Gupta, V., Shantz, S.: Energy Analysis of Public-Key Cryptography for Wireless Sensor Networks. In: The 3rd IEEE International Conference on Pervasive Computing and Communications (2005)
16. Gupta, V., Millard, M., Fung, S., Zhu, Y., Gura, N., Eberle, H., Shantz, S.: Sizzle: A Standards-Based End-to-End Security Architecture for the Embedded Internet. In: The 3rd IEEE International Conference on Pervasive Computing and Communications (2005)

A Trust Approach for Node Cooperation in MANET

Kun Wang and Meng Wu

College of Communication and Information Engineering, Nanjing University of Posts and Telecommunications,
210003 Nanjing, China
skydondon@gmail.com, wum@njupt.edu.cn

Abstract. In MANET, Node misbehavior due to selfish or malicious reasons can significantly degrade the performance of ad hoc networks. To cope with misbehavior in such self-organized networks, an incentive mechanism must be in place. In this study, a trust-based incentive model on a self-policing mechanism is proposed to make collaboration rational for selfish/malicious nodes. This trust system makes them evaluate the trust of their neighbor locally to mitigate contamination by direct observation and the use of second-hand information available. In our approach, every node maintains a trust rating about every node else they care about. Meanwhile, trust fading and redemption are brought about by update and re-establishment of the trust to show robustness. Performance by simulation reveals that in the case of existing malicious nodes, Dynamic Source Routing (DSR) with proposed trust-based node incentive mechanism performs better than DSR in terms of packet successful delivery ratio and mean number of packets dropped.

Keywords: Node cooperation; trust; MANET (Ad Hoc).

1 Introduction

Ad Hoc network [1] is a multi-hop wireless network independently of any base station or fixed infrastructure. In ad hoc network, all networking functions must be performed by the nodes themselves. Thus each node acts not only as a terminal but also a router. Due to lack of routing infrastructure, they have to cooperate to communicate. Nodes are rational, their actions are strictly determined by self interest, and each node is associated with a minimum lifetime constraint. Therefore, misbehavior exists. Misbehavior, which means deviation from regular routing and forwarding, arises for several reasons. Unintentional misbehavior happens when a node is faulty for the linking error or the battery exhausting, while intentional misbehavior can aim at an advantage for the misbehaving node or just constitute vandalism, such as enabling a malicious node to mount an attack or a selfish node to save energy. Malicious nodes are the nodes that join the network with the intent of harming it by causing network partitions, denial of service, etc. The aim of malicious node is to maximize the damage they can cause to the network, while selfish nodes are the nodes that utilize services provided by others but do not reciprocate to preserve their resources. These nodes do not have harmful intentions toward the network, though their actions may adversely affect the performance of the network [2, 3]. The aim of selfish nodes is to maximize

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 481–491, 2007.

the benefits they can get from the network. In game-theoretic terms [4], cooperation in mobile ad hoc networks poses a dilemma [5]. To save battery, bandwidth, and processing power, nodes should not forward packets for others. If this dominant strategy is adopted, however, the outcome isn't a functional network when multi-hop routes are needed, and all nodes are worse off. In this situation, an ideal scheme is needed to give nodes an incentive to cooperate.

Roughly, the proposed incentive approaches can be classified into reputation-based (or detection-based) and market-based (or payment-based) schemes. In reputation-based systems, nodes observe the behavior of other nodes and act accordingly by rewarding cooperative behavior and punishing uncooperative behavior. The typical model of this scheme includes "watchdog" and "pathrater" [2], CONFIDANT [3], CORE [6], etc. Another approach to creating incentives for packet relaying is the market-based approach, as is shown in Nuglets [7], Micro-Payment [8], SPRITE [9], etc. In such systems, nodes receive a micro-payment for every packet that they relay; in return, nodes can use these payments to send their own traffic. In addition, recent incentive schemes for nodes cooperation in multi-hop cellular networks have been researched, such as STUB AD HOC [10], which is beyond the range of this study.

CONFIDANT is a reputation system containing detection, trust evaluation and trust refreshment. It has an initial version with predetermined trust, and then it is improved by an adaptive Bayesian reputation and trust system [11]. The main drawback of CONFIDANT is that trust evaluation only takes a parameter of Bayesian into consideration and doesn't embody in different statistics of network. Additionally, CONFIDANT only adopts periodical fading to prevent normal nodes from refusing cooperation. For malicious nodes, however, it lacks redemption mechanism which is vital to them, for the misbehavior of these nodes may be due to other factors (e.g., linking error, battery exhausting). Aiming at the issue above, we propose a trust-based incentive model on a self-policing mechanism to make nodes evaluate the trust of their neighbor locally. This approach can mitigate contamination by direct observation and the use of second-hand information available. Every node maintains a trust rating about every node else that they care about. Meanwhile, trust fading and redemption are brought about by update and re-establishment of trust to show robustness. What's more, that trust is considered as a metric is a key point in our paper.

The rest of this paper is organized as follows. In Section 2, we present our trust model. The trust evaluation process and performance analysis are stated in section 3 and 4. Finally, section 5 concludes the paper and points to some aspects that will be the subject of the future work.

2 Trust Model

Proposed trust model is based on CONFIDANT [3], and the definitions of different symbols are as follows.

2.1 Definition of $T_{i,j}$

Trust value of N_i to N_j is ：

$$T_{i,j} = a\,F_{i,j} + b\,S_{i,j} \quad (\{T_{i,j}, F_{i,j}, S_{i,j}\} \in (0,1),\ \{a, b\} \in (0,1)) \tag{1}$$

Where

$F_{i,j}$ and $S_{i,j}$ are N_i's trust value to N_j by self-detection and by other nodes' information about N_j, respectively.

a and b are weight value parameters, and the weighty of $F_{i,j}$ and $S_{i,j}$ in computing the trust to N_j can be changed by adjusting a and b.

Besides, trust between N_i and N_j is independent and non-transitive, that is, $T_{i,j}$ doesn't equal to $T_{j,i}$ [12].

2.2 Definition of $F_{i,j}$

$F_{i,j}$ is N_i's trust value to N_j by directly detecting. If N_i is in the radio range of N_j (assuming that N_i is the neighbor of N_j), N_i can obtain $F_{i,j}$ by self-computation. $F_{i,j}$ is defined as:

$$F_{i,j}=\Psi(\delta_1,\ \delta_2,\ \delta_3,\ \delta_4) \tag{2}$$

Where

Ψ is a ratio function of monitored traffic statistics pertaining to traffic volume,

δ_1 is the number of incoming packets on the monitored N_j,

δ_2 is the number of outgoing packets from the monitored N_j,

δ_3 is the number of packets from source N_j to destination N_i,

δ_4 is the number of packets from source N_i to destination N_j [12,13].

According to the parameters above, Ψ is defined as:

$$\Psi=\frac{\text{packets actually forwarded}}{\text{packets to be forwarded}}=\frac{\delta_2-\delta_3}{\delta_1-\delta_4} \tag{3}$$

2.3 Definition of $S_{i,j}$

$S_{i,j}$ is N_i's the trust value to N_j by N_x (N_x is also the neighbor of N_j) monitoring N_j. Initially, when N_i enters into network, it doesn't have the monitoring information of N_j from N_x, at this time, $S_{i,j}$ equals to zero, i.e., $T_{i,j}$ equals to $F_{i,j}$. we define *other* as the set of other nodes:

$$other=\{\ \forall\ N_x\in\ other \Rightarrow \exists\ T_{i,x},\ \ s.\ t.\ T_{i,x}\ \geq\ T_H\ \} \tag{4}$$

Where, T_H is a threshold of trust (see section 3.1 in detail) [12].

Using the weighted average [14] of $T_{i,x}$ over all the nodes in *other*, we define $S_{i,j}$ as:

$$S_{i,j}=\frac{\sum\limits_{x\in Other}(T_{i,x}\times T_{x,j})}{\sum\limits_{x\in Other}T_{i,x}} \tag{5}$$

Where

$T_{i,x}$ on molecular and $T_{x,j}$ are the trust value of N_j on N_x and N_x on N_j, respectively,

$T_{i,x}$ on denominator is a weighty parameter.

3 Trust Evaluation Process

Described trust evaluation process is classified into three phases: initial phase, update phase and re-establish phase [12], as shown in Fig.1. Trust is depicted into three domains: nodes in the "High" domain are highly trusted by their neighbors, and their trust will be utilized to compute $S_{i,j}$; nodes in the "Medium" domain are those with intermediate trust values, and their trust is not utilized in computing $S_{i,j}$; and the nodes in the "Low" domain are those marked as un-trusted malicious nodes.

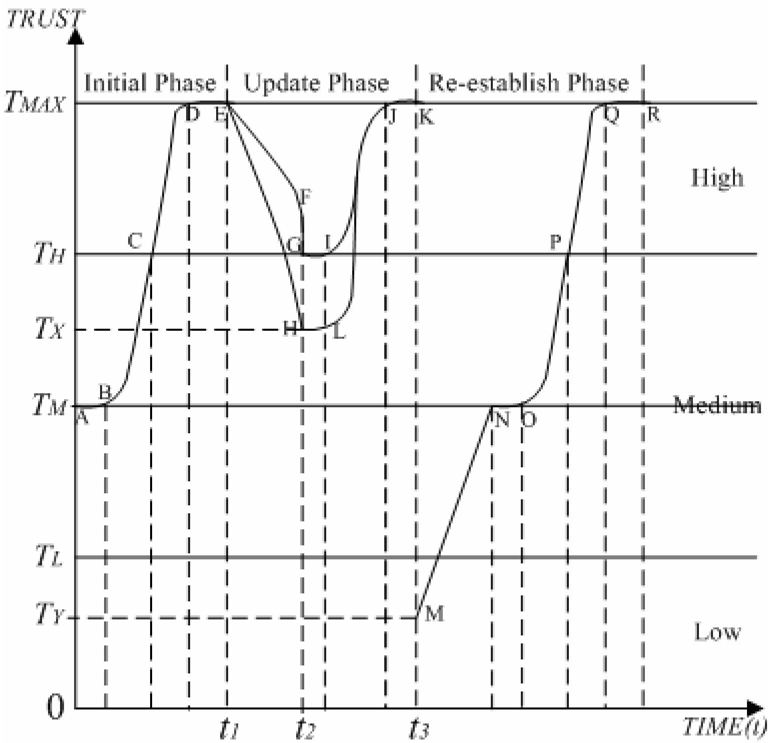


Fig. 1. Trust Evaluation Process

3.1 Trust Initial Phase

When a new node joins a network which already exists, other nodes in this network have no traffic statistics about this new node. As a result, the new node does not have any trust information about its neighbors and vice-versa. We can see in Fig.1, there is a scenario that a new node N_j joins the network. Then N_j is given a trust value T_M, and it starts being monitored by its neighbor N_i, as is shown in AB. At this time, N_i doesn't have the monitoring information of N_j from N_x, so $S_{i,j}$ equals to zero, i.e. $T_{i,j}$ equals to $F_{i,j}$. N_i computes $F_{i,j}$ by equation (2). If N_j is not a malicious node, the trust value of N_i on N_j will reach $T_H$in some time, represented by BC, after which N_i can computes $T_{i,j}$ by equation (1) till the trust value of N_i on N_j reaches T_{MAX}, as the curve CD. As long as N_j doesn't depart from the network, i.e., N_i is all along the neighbor of N_j, the detection of N_i to N_j and trust computation will be carried out continuously. Similarly, if there is no misbehavior on N_j all the time, the trust will keep to T_{MAX}, as the curve DE.

3.2 Trust Update Phase

When N_j left the scope of detection of N_i (i.e., N_j is no longer the neighbor of N_i) at t_1 due to node's mobility and now are back in radio range again, the weighty parameter a in equation (1) will fade during Δt ($\Delta t = t_2 - t_1$). Here, it decays exponentially as:

$$a = \lambda \cdot a \cdot e^{-c \cdot \Delta t} \tag{6}$$

Where

c is a decadent factor,

λ is a constant.

If the trust value fades to the value above T_H at the time t_2, $T_{i,j}$ equals to T_H, as the curve EFG; if it fades to T_X below T_H ($T_L \le T_X \le T_M$), $T_{i,j}$ equals to T_X, as the curve EH. T_H and T_X are the trust value that N_i keeps on N_j after N_j is out of the scope of detection of N_i.

When N_j are back in N_i's radio range again (i.e., N_j becomes the neighbor of N_i again), N_i computes $T_{i,j}$ at T_H or T_X, but not at T_M, as the curve GIJK or HLJK.

3.3 Trust Re-establish Phase

Some nodes will be defined as malicious nodes because of the linking errors or the battery exhausting. Therefore, a redemption mechanism is needed for "malicious" nodes to regain the trust of other nodes. According to the value $T_{i,j}$, N_i won't choose N_j as its forwarding node when N_j, whose trust is in "Low" domain, intends to establish trust with its neighbor N_i. Consequently, the trust value needs increasing periodically to T_M, which is the re-establishment process for nodes' trust, as represented by MNOPQR. The re-establishment function $T_{i,j}'$($T_Y \le T_{i,j}' \le T_M$) is defined as:

$$T_{i,j}' = \begin{cases} T_Y, & t = t_3 \\ T_Y + k \cdot t, & t > t_3 \end{cases} \quad (0 < T_Y \le T_L) \tag{7}$$

Where

k is a incremental slope, which decides the rate of re-establishment process, that is, the redemption rate,

T_Y is the N_i's trust value on N_j at the time t_3.

When $T_{i,j}'$ reaches T_M, N_i computes the trust value on N_j by equation (1).

4 Performance Analysis

In this section, we present the simulation of DSR enhanced with trust model (T-DSR), compared with common DSR. We implemented our protocol using NS 2 (version

2.29) with CMU wireless extension [15]. The common parameters for all the simulation runs are listed in Tab.1.

Table 1. Simulation Settings

Parameters	Value	Parameters	Value
Simulation Area	1000units×1000units	MAC Protocol	IEEE 802.11
Transmission Range	200units	Linking Capacity	2Mbps
Traffic Source	CBR	Packer Size	64 bytes
Mobility Model	Random Way Point	Simulation Duration	800s
Sending rate	0.5 pkt/s	Node Speed	[0,20]units/s

Given that receiving ratio and packet dropping ratio can reflect nodes cooperation, we utilized this two parameters to illustrate the performance of T-DSR.

1) Packet successful delivery ratio (PDR),

$$\text{PDR}=\frac{\text{data packets delivered to the destination}}{\text{data packets generated by the CBR sources}}$$

2) Mean Number of Packets Drooped (MNPD),

$$\text{MNPD}=\frac{1}{n}\sum_{i=0}^{n-1}(\text{data packets generated -data packets received})$$

We firstly set different number of malicious nodes in network with different total number of nodes, and analyzed what impact the node mobility (change of pause time) on PDR and MNPD. Subsequently, we considered the main parameters *a*, *b*, *c* and *k* （λ is set to 1） to analyze the effects on above two parameters. The results are demonstrated in Fig. 2 to Fig. 8.

We fixed the parameters $a=b=0.5$, $c=1$, $k=0.005$. The network size is set to 60, 30 and 15, in which there are 20, 10 and 5 malicious nodes respectively. The MNPD of T-DSR and DSR varying the pause time are shown in Fig.2. We can see that

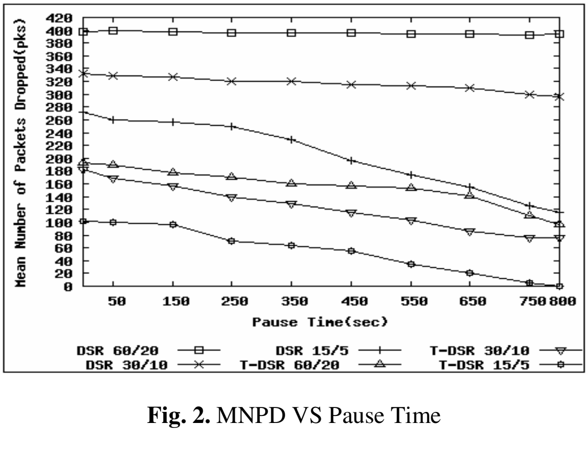


Fig. 2. MNPD VS Pause Time

MNPD of DSR is much greater than that of T-DSR, which means T-DSR effectively stimulates the nodes cooperation. On the other hand, pause time has little influence on MNPD, especially in DSR. However, with the increment of node mobility, T-DSR is a little more affected, since malicious nodes have more opportunities to be forwarding nodes, which enlarge the MNPD.

When pause time is set to zero and network size to 60, the MNPD and PDR of two compared protocols with the change of number of malicious nodes are depicted in Fig.3 and Fig.4, respectively. As is shown in Fig.3, when the number of malicious nodes increases to 10, MNPD of T-DSR and DSR augments rapidly, after which MNPD of T-DSR increases slowly while that of DSR keeps flat. It is indicated that T-DSR can allow the existence of small number of malicious nodes. In Fig.4, at the beginning, PDR of DSR descends speedily, during which time PDR of T-DSR is much greater than that of DSR. While the number of malicious nodes increases to 25, PDR of T-DSR descends speedily as well, and at the point of 50, PDR of T-DSR and DSR are almost consistent.

When network size is fixed at 60, in which the number of malicious nodes is 20, PDR of T-DSR and DSR varying the pause time are shown in Fig.5, from which we

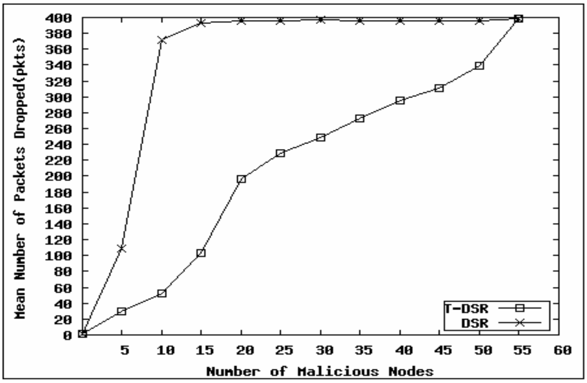


Fig. 3. MNPD VS Number of Malicious Nodes

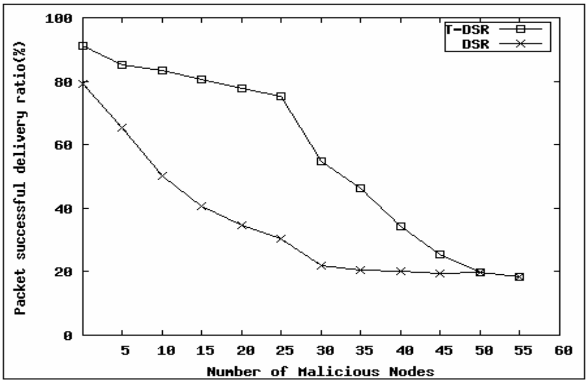


Fig. 4. PDR VS Number of Malicious Nodes

can see that PDR of T-DSR is much greater than that of DSR. While PDR of DSR without malicious nodes is almost the same as that of T-DSR with 20 malicious nodes, by the reason that besides malicious nodes' denying of services which lead to packet dropping, other factors, such as linking error, can result in the descent of PDR.

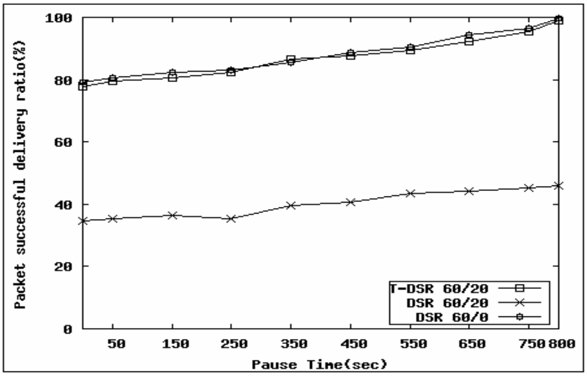


Fig. 5. PDR VS Pause Time

Finally, we gave thought to the influence on PDR by the main parameters *a*, *b*, *c* and *k* in T-DSR. As described above, we fixed network size at 60, and the number of malicious nodes at 20, with the variant pause time. Results are shown in Fig. 6, 7 and 8. It is can be seen in Fig.6 that the monitoring information (second-hand information) other nodes provide is in favor of discovering malicious nodes more quickly, which makes cooperation between nodes intensified. If second-hand information is out of use, that is $a=1$, $b=0$, PDR will obviously decline. In Fig. 7 we can see that if there is no fading in nodes' trust, i.e., $c=0$, nodes cooperation will inevitably weaken, since those nodes whose trust reach top will alternatively refuse to forward packets. Fig.8 denotes that the redemption rate (value of *k*) of malicious nodes will impact on nodes cooperation as well. Providing

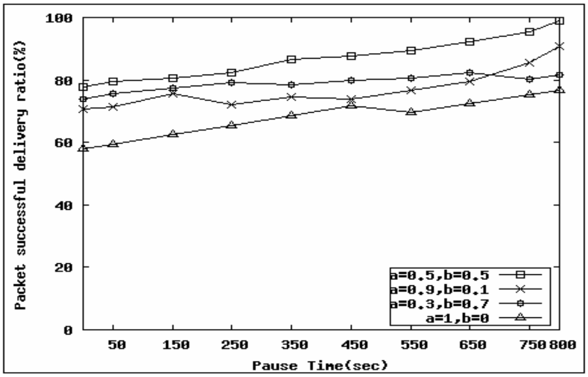


Fig. 6. $c=1$, $k=0.005$, PDR VS Pause Time

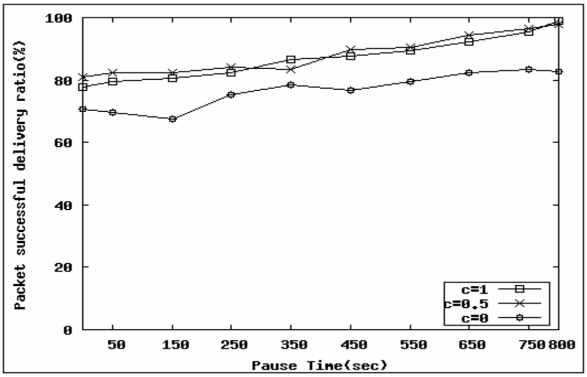


Fig. 7. *a*=*b*=0.5, *k*= 0.005, PDR VS Pause Time

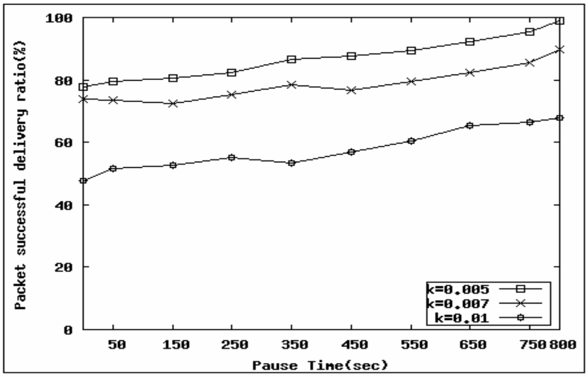


Fig. 8. *a*=*b*=0.5, *c*= 1, PDR VS Pause Time

that redemption rate is excessively fast (see *k* =0. 01), likewise, malicious nodes can rejoin in network quickly, which will be bound to affect PDR, while slow redemption rate can make network operate in safer environment with fewer malicious nodes.

5 Conclusions

By the analysis of CONFIDANT, a trust-based incentive model on a self-policing mechanism is proposed to make collaboration rational for nodes and to allow the existence of selfish/malicious nodes. This trust system in all nodes makes them evaluate the trust of their neighbor locally to mitigate contamination by direct observation and use of second-hand information available. In our approach, every node maintains a trust rating about every node else that they care about. Meanwhile, trust fading and redemption are brought about by update and re-establishment of trust to show robustness. Performance by simulation reveals that in the case of existing malicious nodes, DSR with proposed trust-based node incentive mechanism performs better than DSR in terms of packet successful delivery ratio and mean number of packets dropped. While security problem of this mechanism needs further considering, for reputation-based scheme suffers some problems. Firstly, reputation systems can either rely exclusively on their own observations or also consider information obtained by others. However, secondhand information can be spurious, which raises the questions of how to incorporate it in a safe way and whether to propagate it. Secondly, when a node has been isolated, it can no longer be observed. If node's misbehavior is temporary, a redemption mechanism ensures that it can come back to the network. However, preventing recidivists from exploiting a redemption mechanism is desirable. Thirdly, when the number of misbehaving nodes is too large, some additional problems will come forth, such as brainwashing, intoxication and identity spoofing [16]. Hence, security issues on this trust model deserve to be attached importance to, which is also our following target.

Acknowledgments. This work is supported by National "863" High Technology Research and Development Program Foundation of China (No.2006AA01Z208); Six Talented Eminence Foundation of Jiangsu Province, China; Natural Science Foundation of Jiangsu Province, China (No.BK2007236).

References

1. David, R., Ignas, G.N.: Ad Hoc networking in future wireless communications. Computer Communications 26, 36–40 (2003)
2. Marti, S., Giuli, T.J., Lai, K.: Mitigating routing misbehavior in mobile MANET networks. In: 6th International Conference of Mobile Computer Networks, Boston, MA, pp. 255–265 (2000)
3. Buchegger, S., Le Boudec, J.Y.: Performance Analysis of the CONFIDANT Protocol (Cooperation of Nodes-Fairness in Dynamic Ad-Hoc NeTworks). In: 3rd ACM International Symposium of Mobile MANET Networking and Computing, pp. 80–91 (2002)
4. Fudenbery, D., Tirole, J.: Game Theory. Cambridge, MA (1991)
5. Rapaport, A., Chammah, A.M.: The Prisoner's Dilemma: A study in conflict and cooperation. University of Michigan Press, Ann Arbor, MI (1965)
6. Michiardi, P., Molva, R.: Core: A COllaborative REputation mechanism to enforce node cooperation in Mobile Ad Hoc Networks. In: IFIP - Communication and Multimedia Security Conference, pp.107–121 (2002)
7. Buttyan, L., Hubaux, J.P.: Nuglets: a virtual currency to stimulate cooperation in self-organized MANET networks. Technical Report DSC/2001/001, Swiss Federal Institute of Technology, Lausanne (2001)
8. Jakobsson, M., Hubaux, J.P., Buttyan, L.: A Micro-Payment Scheme Encouraging Collaboration in Multi-Hop Cellular Networks. In: 7th International Financial Cryptography Conference, Berlin, Germany, pp. 15–33 (2003)
9. Zhong, S., Chen, J., Yan, Y.R.: Sprite: a simple, cheat-proof, credit-based system for mobile ad-hoc networks. In: IEEE Conference on Computer Communications, San Francisco, CA, pp. 1987–1997 (2003)
10. Lamparter, B., Paul, K., Westhoff, D.: Charging Support for MANET Stub Networks. Computer Communications 26, 1504–1514 (2003)

11. Buchegger, S., Le Boudec, J.Y.: A Robust Reputation System for Peer-to-Peer and Mobile Ad Hoc Networks. In: 2nd Workshop on the Economics of Peer-to-Peer Systems, Harvard Univ., Cambridge, MA (2004)
12. Mohit, V., Murtuza, J., Madhusudhanan, C.: Quantifying Trust in Mobile Ad-Hoc Networks. In: IEEE International Conference on Integration of Knowledge Intensive Multi-Agent Systems, Waltham, MA, USA, pp. 65–70 (2005)
13. Huang, Y., Lee, W.: A Cooperative Intrusion Detection System for Ad Hoc Networks. In: ACM Workshop on Security of Ad Hoc and Sensor Networks, Fairfax VA, pp. 135–147 (2003)
14. The weighted average definition, http://www.mathwords.com/w/weighted_average. htm
15. The Vint Project. The ns-2 network simulator, http://www.isi.edu/nanam/ns
16. Buchegger, S., Le Boudec, J.Y.: Self-policing Mobile Ad Hoc Networks by Reputation Systems. IEEE Communications, 101–107 (2005)

A Clustering Algorithm Based on LBG and VQ in Mobile Ad Hoc Network

Xiaolei Wang*, Yunfei Guo, and Jiang Ji

National Digital Switching System Engineering & Technological Center, zhengzhou, P.R. China
shelraywang@gmail.com

Abstract. Apply the theory of vector quantization and LBG algorithm in the Information Theory and Data Compress,this paper present a one by one relationship between the cluster and cell,the cluster head and code vector,the node and vector,turn the process of clustering into the process of cell segmenting.Then design a new clustering algorithm aiming high stability in cluster structure and good performance in load balancing of cluster head.It has been proved that this algorithm has good performance in load balancing and the structure of the cluster has high stability after the simulation.

1 Introduction

Mobile Ad Hoc Network (MANET) is a new type of dynamic network which constituted of mobile nodes and independent of basic establishment.It adopts non-central distributed control.Compared with the cellular communication and wireless LAN which needs central control,the mobile Ad Hoc has high quality in organizing,robust and invulnerability[1,2].It has been used widely in the building of military information system,also in the universal exigency salvation and other places where need to set up temporarily network.

The clustering algorithm is an important way of actualizing hierarchical routing in mobile Ad Hoc network.The clustering is a method to divide the nodes in hierarchy in the network,some of the nodes constitute a cluster.The members in a cluster can be divided into cluster heads,cluster members and gateways.The cluster heads charged the communication between the cluster members and the gateways charged the communication between different clusters.There are several standards in the following which used to estimate the merits of the clustering algorithm:

*(1) The stability of the cluster structure:*The stability of the cluster structure is measured by the changing frequency of the cluster head.In one unit time,the less number of the changed cluster heads,the high stability of the cluster structure,and the less cost of the communication.

* XiaoleiWang(1982-), male,postgraduate student, The research domain is mobile communication system; Yunfei Guo(1963-), male, professor, doctor mentor, The research domain is mobile communication system;Jiang Ji(1982-), male, postgraduate student, The research domain is mobile communication system.

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 492–499, 2007.

*(2)The number of cluster heads:*In the case of link allowed,the algorithm with less cluster heads can effectively improve the utilization rate of the link.Because of the resource constraint of the nodes,the number of the cluster members should be restricted when limiting the number of the cluster heads.

*(3)Load balancing:*The maintenance of the cluster structure and the routing between different clusters need to cost some resource,so it's not reasonable that some cluster heads over load while another cluster heads have nothing to do.The load of the cluster head depend on the number of cluster members which it can afford.In order to test the load balancing of the cluster head,literature[3] brought forward the concept of the load balancing:

$$LBF = \frac{n_i}{\sum_i (x_i - u)^2} \quad . \tag{1}$$

u is the average number of cluster members that the cluster head affords,$u = (N - n_c)/n_c$,$N$is the number of nodes in the network, $n_c$is the number of the cluster heads,$x_i$is the number of the cluster members of cluster head.

Recently there are many clustering algorithms,for example the smallest ID algorithm,the largest linking algorithm,and the passive clustering algorithm[1].In the smallest ID algorithm every node has an exclusive ID,and the adjacent nodes periodic exchange state message.The node with the smallest ID becomes the cluster head.Simple is the most excellence of this algorithm,but it lacks justice in the Ad Hoc network with energy limited.The standard of choosing cluster heads in the largest linking algorithm is the linking which means the number of neighbors of one node.Because the number of cluster heads is less and there is no constraint to the number of nodes in the cluster in this algorithm,it can't be applied to the Ad Hoc network which needs to limit the ability of the nodes.The passive clustering algorithm doesn't use special control message during the clustering.It obtains clustering message from natural data package.This algorithm has the excellence of saving bandwidth and it is considered as a clustering technology with good development.

Take the optimizing of load balancing and the stability of cluster structure as the goal.This paper designed a new clustering algorithm by Combining the theory of vector quantization and LBG algorithm.

2 Vector Quantization[4]

In K dimensional space R^K,divide R^K into J disjoint subspace $R_1, R_2, \ldots, R_J$,which satisfy:

$$\begin{cases} R_1 \bigcup R_2 \bigcup, \ldots R_J = R^K \\ R_i \bigcap R_j = \phi, \ \ when i \neq j \end{cases} \quad . \tag{2}$$

In each subspace R_i find a represent vector Y_i and these vectors compose a gather $Y = \{Y_1, Y_2, \ldots, Y_J\}$.$Y$ is called codebook,Y_i is called code vector,the number of code vector J is called codebook length.

From another vector's gather $X = \{X_1, X_2, \ldots, X_N\}$ arbitrary extractive a vector $X_j \in R^K$,when input X_j,VQ firstly make a judgment about which subspace it is belonged,then output the represent vector Y_i of subspace R_i.In a word the process of VQ is a process of using Y_i to express X_j,and that is:

$$Y_i = Q\left(X_j\right), 1 \le i \le J, 1 \le j \le N \ . \tag{3}$$

Q is called quantizer.

The best quantizer must satisfy two necessary conditions:

(1)Voronoi segmentation—the segmentation of R^K should satisfy:

$$R_i = \left\{X \in R^K : d\left(X, Y_i\right) \le d\left(X, Y_k\right), i \ne k\right\}, i, j = 1, 2, \ldots J \ . \tag{4}$$

$d\left(X_j, Y_i\right), j = 1, 2, \ldots N; i = 1, 2, \ldots J$ is the average distortion when use Y_i to express X_j.R_i is called Voronoi cell.Any vector in the cell can be expressed by the code vectorY_i.

(2)Centroid condition—When the Voronoi cell segmentation is complete,the centroid of cell R_i become the code vector Y_i,that is:

$$Y_i = E\left[X \mid X \in R_i\right] \ . \tag{5}$$

In a word,the formation of best quantizer is a process of iterative optimizing to cell and code vector.It is a process of finding the best cell and the best code vector.

The initial structure of mobile Ad Hoc network was 2D plane[5,6]which meant all nodes in the network have the same energy and bandwidth resource,and charged with two functions:terminals and routing.The state and control message can be exchanged between adjacent nodes in real time.Assume that the dynamic characteristic of the nodes change slowly which means there will be no node join or leave the network in a short period.We can take the proximity of location as the clustering standards, in the condition of ignoring the communication cost of the nodes and the link state.

So,the clustering problem of mobile Ad Hoc network has become a two-dimensional problem of VQ.The process of clustering is the segmentation of R^K.The Voronoi cell is the cluster,and the code vector is the cluster head,the length of codebook is the number of cluster,the number of vector is the number of node.

3 The Clustering Algorithm Based on LBG and VQ

In order to meet the requirements of load balancing,the size of cluster should be limited in the process of clustering.It is often restricted based on the energy cost, bandwidth and link state of nodes[7,8].From the two necessary conditions which the best quantizer must be satisfied,we can see that in order to seek the suitable cell firstly need to seek the suitable codebook.The generation of the best codebook based on the principle of minimum distortion.That's to minimize the distortion $\overline{D}$ when use Y_i to express X_j.

$$\overline{D} = \min\left\{E\left[d\left(X, Y\right)\right]\right\} \ . \tag{6}$$

Linde,Buzo and Gray put the most optimized scalar quantization M-L algorithm extend to the Multi-dimensional space,which was usually called LBG algorithm.Because of the tightly characteristic of theory and the convenience of actualizing,the LBG algorithm has been widely used,especially used in the area of mobile communication.LBG algorithm can be used to optimize codebook,and the initial codebook can be generated by using segmentation method[4].

3.1 The Clustering Process of 2D VQ

Take the nodes in the mobile Ad Hoc network as a 2D vector's gather $X = \{X_1, X_2, \ldots, X_N\}$.Take the proximity of location as the clustering standards. Set the max distance between any two nodes in a cluster is L,and take this as the convergence condition of the algorithm.Design the clustering process of 2D VQ as follow:

*Step1:*Find the centroid Y_1^0 of all the nodes in the vector's gather X.

$$Y_1^0 = \frac{1}{N}\sum_{i=1}^{N} X_i \ . \tag{7}$$

*Step2:*The first segmentation generates initial codebook with the length of 2.

$$Y_2^0 = \{Y_1, Y_2\} \ . \tag{8}$$

$Y_1 = Y_1^0, Y_2 = a \cdot Y_1^0$,and a is a constant obtained by experience. Using the LBG algorithm make iterations for m times to the initial codebook, and find the best codebook Y_2^m:

$$Y_2^m = \{Y_1^m, Y_2^m\} \ . \tag{9}$$

At the same time, two best cells $R_1, R_2$are segmented, that means two clusters are generated.

*Step3:*The $J - 1$ times segmentation generates initial codebook Y_J^0 with the length of J.

$$Y_J^0 = \{Y_1, Y_2, \ldots Y_J\} \ . \tag{10}$$

$Y_1 = Y_1^m, Y_2 = Y_2^m, \ldots Y_J = a \cdot Y_{J-1}^m$, and a is a constant obtained by experience.Using the LBG algorithm make iterations for m times to initial codebook, and find the best codebook Y_J^m:

$$Y_J^m = \{Y_1^m, Y_2^m, \ldots Y_J^m\} \ . \tag{11}$$

At the same time,J cells $R_1, R_2, \ldots R_J$ are segmented, that means J clusters are generated.

After $J-1$ times segmentation, if generate empty cell R_i $(i = 1, 2, \ldots J)$, then cut off the empty cell, and find the biggest cell R_j.Divide R_j into two cells and take them to replace the empty cell, which must satisfy:$Y_1 = Y_1^m, \ldots Y_j = Y_j^m, \ldots Y_i = b \cdot Y_j^m, \ldots Y_J = Y_{J-1}^m$,$b$ is a constant obtained by experience.

Step4:In the J cells which have been generated, if the max distance of any two nodes is shorter than L, then stop the segmentation and J clusters are generated.

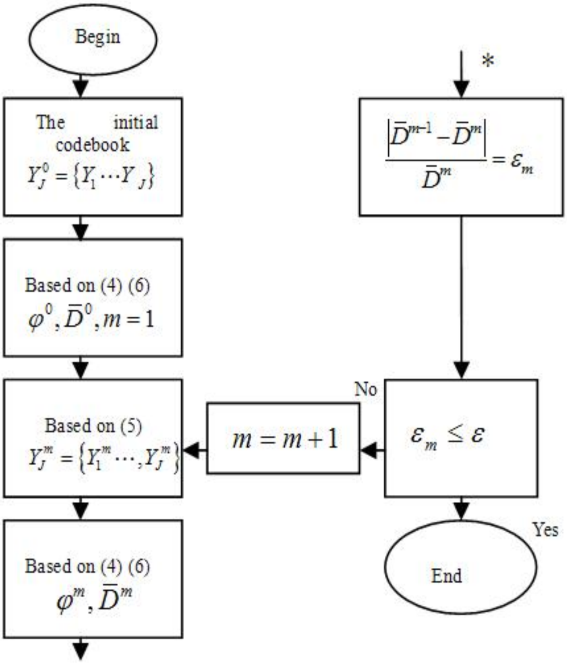

Begin
The initial codebook
$Y_J^0=\{Y_1\cdots Y_J\}$
Based on (4) (6)
$\varphi^0,\bar{D}^0,m=1$
Based on (5)
$Y_J^m=\{Y_1^m\cdots,Y_J^m\}$
Based on (4) (6)
$\varphi^m,\bar{D}^m$
*
$\frac{|\bar{D}^{m-1}-\bar{D}^m|}{\bar{D}^m}=\varepsilon_m$
No
$m=m+1$
$\varepsilon_m\leq\varepsilon$
Yes
End

Fig. 1. Flowchart of LBG algorithm

3.2 The Process of LBG Algorithm

The length of codebook J is confirmed by the clustering process of VQ. The distortion control bound ϵ is confirmed by experience and requirements of the network.$\varphi = \{R_i, i = 1, 2, \dots J\}$is a set of cells, and $\varphi^i, i = 1, 2, \dots m$ is the set of cells that after i times iterative optimization through LBG algorithm.

4 The Simulation and Test of This Algorithm

4.1 The Simulation of Clustering Result

In order to show the currency of this algorithm,randomly create 926 nodes of mobile Ad Hoc network,and randomly distribute these nodes on a square area which cavern an ellipse in the center.The four vertexes of this square in the 2D coordinate system are:$(0, 0), (0, 1000), (1000, 0), (1000, 1000)$,the elliptical center is$(550, 550)$,the long axes is 410,the short axes is 210,and the long axes has a 30 degree angle with the X axes, see Fig.2(a).

Assume L the max distance between any two nodes in the cluster is 100, the distortion control bound ϵ is 0.01, a is determined accord to the experience in every segmentation of cells.

For the sake of observing the result of clustering, after the clustering simulation in MATLAB,set the code vector as the centre of a circle,set the farthest

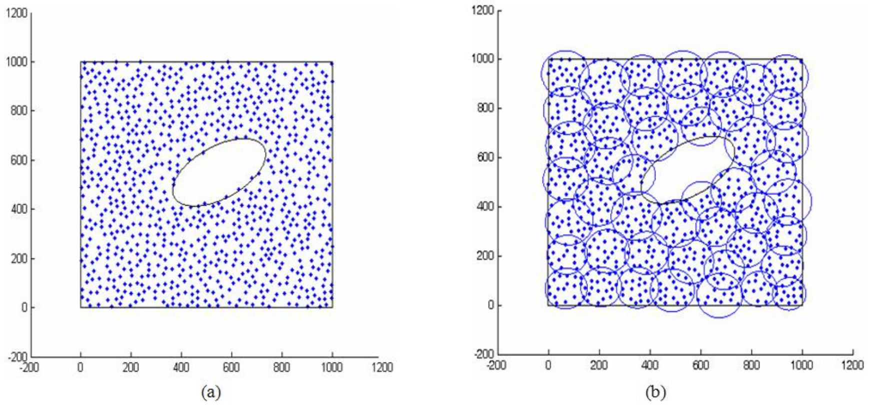


Fig. 2. The simulation of clustering algorithm based on LBG and VQ.(a)shows the nodes distribution before clustering.(b)shows the nodes distribution after clustering.

node away from the code vector as radius, draw a round,and call this round as initial round.The initial round covers all the nodes in current cluster and some nodes in another cluster.It is necessary to confirm the gateways for assuring the communication between different clusters.By dynamic adjusting the location of initial round,make sure the common area of the adjacent rounds not larger than the bigger round in 5 percent[9].After the adjustment we can confirm the gateways,the clustering result of this algorithm is shown in Fig.2(b).

From the Fig.2(b) we can see that the code vector become the cluster head,the nodes in the round become the cluster members,and the nodes in the common area of the adjacent rounds become the gateways.

After the clustering algorithm,there are 46 clusters generated,and also generate 46 cluster heads.We can calculate the load balancing is: 1.02 accord to (1).Then we can confirm that this algorithm has good performance in the load balancing.

4.2 The Stability Test of Cluster Structure

A stable structure of the cluster can decrease the cost of communication and energy,and it won't rebuild the cluster easily just because of some nodes join or leave the network.For the sake of testing the stability of the cluster structure,we need to make a statistic of the changed cluster heads in one unit time according to its concept.

In the area which has been clustered successfully,delete some nodes randomly,and then use the clustering algorithm to cluster again.Account the number of the changed cluster heads.Once there are many cluster heads changed,that means the changing frequency of cluster head is high,and the stability of the cluster structure is low,also the probability of communication interrupt is high.

We can see that the probability of communication interrupt in the network has a direct ratio with the number of cluster heads which have been changed[10,11].We can use the probability of communication interrupt during the section $(0, 1)$ to score the stability of cluster structure.The test of stability of cluster structure is showed in Fig.3:

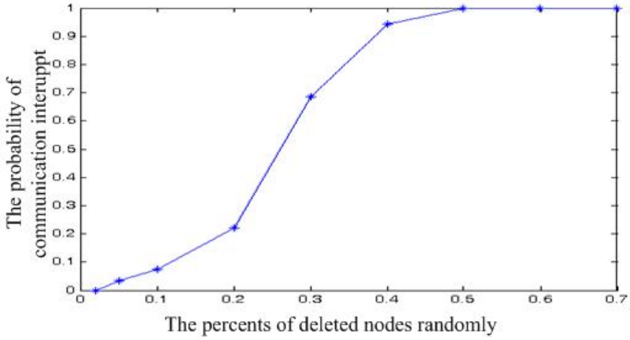


Fig. 3. The test pattern of stability of cluster structure

From the Fig.3 we can see that along with the increasing of the number of deleted nodes,the probability of communication interrupt rising,and the stability becoming worse.When deleted 50 percent of nodes,all the cluster heads in the network changed,and the network communication interrupted with the probability 1;when deleted nodes under 2 percent,there is no cluster heads in the network changed,and the network communication won't be interrupted.

After the clustering with this algorithm, in the condition of delete little nodes,the cluster structure still in high stability.

5 Epilogue

At present there are many algorithms about clustering.This paper apply the VQ theory and LBG algorithm in the Information Theory and Data Compress to the clustering problem in mobile Ad Hoc network,and offer a new method in the research of clustering algorithm.There are several excellences of this algorithm:It can confirm the cluster heads and cluster members rapidly;The stability of cluster structure and load balancing is well;It is simple and easy.This algorithm belongs to centralize algorithm,the nodes must be able to exchange the state message between each other in real time,and the algorithm doesn't consider the bandwidth of the cluster head.The clustering problem in mobile Ad hoc network and wireless sensor network still locate in researching phase.

In the future there will be several improvements:(1) Consider the diversity of distributing area.There will be clustering problem in 3D space network;(2)Decrease the cost of communication, increase the stability of the cluster structure, and decrease the chance of cluster rebuilding;(3)Security problem of the algorithm. Prevent the false message sending by the hostile nodes.

References

1. Hu, G., Jiang, J., Hu, G.: A Survey on Clustering Algorithm in MANETS. Computer Engineering&Science 127(1) (2005)
2. li, L., Dong, S., Wen, X.: Clustering Algorithms in Wireless Sensor Network. In: Wirless Communication Network (March 2003)
3. Chatterjee, M., Das, S.K., Turgut, D.: WCA: A Weighted Clustering Algorithm for Mobile Ad Hoc Nerwork[J]. Journal of Clustering Computing IEEE 5(2), 193–204 (2002)
4. Wu, L.: Data Compress[M],Beijing, Electronic Industry Publishing Company (2000)
5. Zou, Y., Chakrabarty, K.: Sensor Deployment and Target Localization Based on Virtual Forces[J]. In: IEEE INFORCOM 2003 (2003)
6. Poduri, S., Sukhatme, G.S.: Constrained Coverage for Mobile Sensor Networks[J]. In: IEEE International Conference on Robotics and Automation, pp. 165–172 (April 26-May 1, 2004)
7. Ghiasi, S., Srivastava, A., Yang, X.: Majid Sarrafzadeh, Optimal Energy Aware Clustering in Sensor Networks. Computer Science Department, University of California at Los Angeles (2002)
8. Li, Z., Guo, Z., Yu, Z., Zhao, Y.: Energy optimized strategy based on clustering arithmetic, Computer Engineering and Design (2006)
9. Feng, Y., Wang, G., Liu, Z., Jiang, Q.: A Clustering Algorithm Applied to the Management of Mobile Ad Hoc Network. Journal of Software 14(1), 132–138 (2003)
10. Cheng, W., Zheng, J., Sheng, L.: A Clustering Model for Mobile Ad Hoc Network. Journal of Computer Research and Development (2004)
11. Liu, C.: The method of mathematic modeling[M]. Higher Education Publishing Company, Beijing (2002)

Cooperation Transmission Without Perfect Synchronization for Wireless Sensor Networks Using Signal Space Diversity and STBC

Li-li Guo and Dian-wu Yue

College of Information Engineering, Dalian Maritime University,
116026 Dalian, Liaoning Province, China
lengyue_lily@126.com, dianwuyue@yahoo.com

Abstract. Cooperative transmission breaks away from traditional transmission which only has single transmitter and single receiver in wireless networks, which enable single antenna mobiles in a multi-user environment to share their antennas and generate a virtual multiple-antenna transmitter in order to achieve transmit diversity. Space-time block coding (STBC) is an attractive transmission diversity technique because of its linear complexity. Furthermore, signal space diversity can increase the diversity order through interleaving over coordinates and choosing a proper signal constellation. In this paper, cooperative transmissions for wireless sensor networks are considered. In order to improve the system performance, a new cooperative transmission scheme without perfect synchronization using both signal space diversity and STBC is proposed. It is shown by simulation that compared to the traditional transmission scheme with signal space diversity and the STBC-encoded cooperative transmission scheme without signal space diversity, the new scheme can provide further performance improvement by increasing the diversity order, and can save more energy in wireless sensor networks.

Keywords: Cooperative transmission, wireless sensor network, signal-space diversity, space-time block code.

1 Introduction

In wireless sensor networks, energy efficiency is a dominating design criterion. Since wireless sensor network is a dynamic multi-hop network which comprises many deployed sensors, each sensor is powered by battery and supposed to work without replenished for a relatively long time after deployment, it is very important to improve transmission power efficiency and reliability, especially considering the severe channel fading and node failure in hostile environment [1,2].

Space-time block codes (STBC) and processing are helpful for enhancing transmission energy efficiency if antenna arrays are available [3]. Because of affordable complexity, Space-time block codes have attracted great attention for wireless sensor networks. However, each sensor is limited by size or hardware complexity to one antenna.

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 500–510, 2007.

Cooperative communication has been proposed that enables single antenna mobiles in a multi-user environment to share their antennas and generate a virtual multiple-antenna transmitter that allows them to achieve transmit diversity [1]. Therefore, for each sensor where antenna arrays are not available, STBC may be used with cooperative transmission schemes. In this paper, we utilize the simplest Alamouti's code which obtains full diversity and full rate.

Besides of antenna arrays, another challenge for wireless sensor networks is synchronization problem among transmission. So far, many researches on cooperative transmission assume perfect synchronization among cooperative users. As a matter of fact, it is difficult, and in most cases impossible, to achieve perfect synchronization among cooperative transmitters to make them identical in carrier frequency, carrier phase, symbol timing and timing phase. This is even more a reality when low-cost, small-sized transmitters are used, such as sensors [4]. Synchronization is difficult because parameters of electronic components may be drifting and because making handshaking among transmitters is as infrequently as possible to save energy. The lack of perfect delay synchronization brings another side effect, i.e., channels become dispersive [5]. [6] addresses delay asynchronism, it is for two transmitters only without considering timing/frequency asynchronism nor dispersive channel.

An alternative scheme to increase diversity order involves interleaving over coordinates and choosing a proper signal constellation [7]. This technique is also known as signal-space diversity or modulation diversity, and is explored by Boutros [8]. The diversity order of a multidimensional signal set is increased to the minimum number of distinct components between any two constellation points, namely, the diversity order is the minimum Hamming distance between any two coordinate vectors of constellation points. An attractive feature of signal space diversity over other schemes is that the rotated signal set has exactly the same performance of the non rotated one when used over a pure AWGN channel [9-13]. Also, the realization of signal-space diversity is comparatively simple for practical system, and only small complexity is added to system transmitter and receiver.

Furthermore, [14] also points out that general linear transformations of information symbols for QOSTBC could have both full diversity and real symbol pair-wise ML decoding. They present the optimal transformation matrices, (especially constellation rotations) of information symbols for QOSTBC with real symbol pairwise ML decoding such that the optimal diversity product is achieved for square QAM. Therefore, utilizing constellation rotations also could decrease decoding complexity, as well as increase diversity order.

In this paper, we investigate cooperative transmission using STBC and signal-space diversity only considering delay asynchronism but without considering dispersive channel. Our numerical results reveal that the new transmission scheme has better system performance and outperforms other schemes by achieving lower SER and BER, for example only single-transmission and only STBC with or without signal-space diversity.

The remainder of this paper is organized as follows. Section 2 introduces the cooperative transmission scheme with STBC without perfect synchronization. In section 3, we describe the new cooperative transmission scheme with signal space diversity. We analysis system performance in section 4, and provide numerical results. Section 5 concludes the paper with the list of our main contributions.

2 Cooperative Transmission Scheme without Perfect Synchronization

The cooperative transmission scheme to be considered in this paper is shown in Fig.1. We consider a wireless sensor network where a source sensor needs to transmit their data to a remote data collector through a multi-hop wireless network. We consider the hop i, assume J nodes ($j=1,\ldots,J$) in the hop i, namely J sensors receive data packets from the previous hop. For traditional single-transmission, in the hop i, only one of J nodes is chosen to transmit the data packet to the next hop $i+1$. However, for the cooperative transmission scheme, in the hop i, two of the J nodes are chosen to transmit in a cooperative manner with STBC encoding.

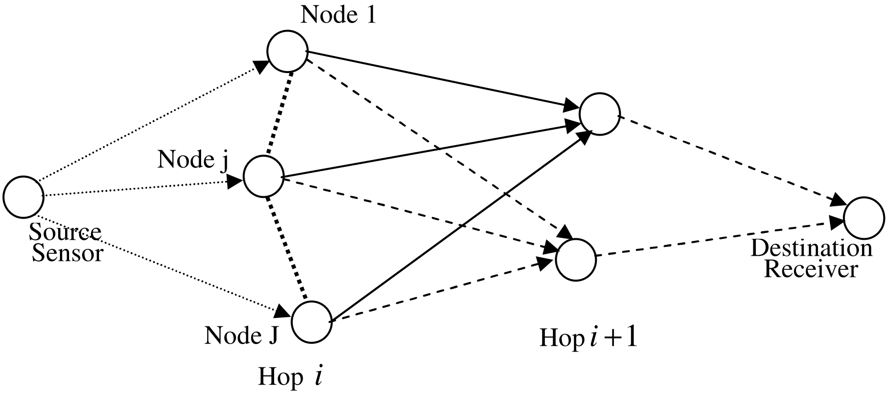


Fig. 1. Multi-hop wireless sensor network with cooperative transmissions

We know that Alamouti's STBC scheme which requires perfect synchronization among the cooperative transmitters is equivalent SISO model, and could achieve the optimal diversity gain. However, that is just the ideal case. In practice, perfect synchronization is very hard, even impossible to realize in wireless sensor networks because low cost implementation of the sensors may make their timing and frequency slightly different, hence may cause mismatch in the long run. The major synchronization problem is represented in the delays of their signals when reaching at the receiver. Propagation delays are usually unknown to the transmitters, while their transmission time may also be different.

2.1 Cooperative Transmission Scheme with Delay Asynchronism

Without loss of generality, considering hop i, assume that sensor 1 and 2 have both received two data packets S1: $\{s_1(1),\cdots s_1(N)\}$ and S2: $\{s_2(1),\cdots s_2(N)\}$ from the previous hop i-1, where $s_i(n), i=1,2$, are uniformly distributed symbols with zero mean and variance σ_s^2. In symbol period 1, sensor1 transmits S1 while sensor 2 transmits S2 simultaneously to the hop $i+1$. Then in symbol period 2, sensor 1 transmits $\{-s_2^*(N),\cdots,-s_2^*(1)\}$ and sensor 2 transmits $\{s_1^*(N),\cdots,s_1^*(1)\}$ to the hop $i+1$, where $(\bullet)^*$ denotes complex conjugate. Considering delay asynchronism, assume transmission delay of sensor 1 is d_1 and delay of sensor 2 is d_2, namely relative delay between sensor 1 and sensor 2 is $|d_1-d_2|$. For fading channels, we consider nodes 1 and 2 transmit data packets to the next hop through Rayleigh flat-fading with coefficients h_1 and h_2 respectively and keep constant over the two symbol periods with changing randomly subsequently. Note the sensor failure probability is included in random fading. The signals received at the receiver over the two consecutive periods are $x_1(n)$ and $x_2(n)$, respectively. Note N samples per symbol period, and $1\le n\le N$.

$$x_1(n)=[h_1 \quad h_2]\bullet\begin{bmatrix} s_1(n-d_1) \\ s_2(n-d_2) \end{bmatrix}+v_1(n) \ . \tag{1}$$

$$x_2(n)=[h_1 \quad h_2]\bullet\begin{bmatrix} -s_2^*(N-n+d_1) \\ s_1^*(N-n+d_2) \end{bmatrix}+v_2(n) \ . \tag{2}$$

Where $v_i(n), i=1,2$ are $1\times N$ noise vectors whose elements are uncorrelated, and are zero mean complex Gaussian random variable with variance σ^2 . $s_1(n)=s_2(n)=0 \quad if\ n<1 \quad or \quad n>N$. Note sample vector $\mathbf{x}(n)=\begin{bmatrix} x_1^*(n) \\ x_2(N-n+d_1+d_2) \end{bmatrix}^H$, where $(\bullet)^H$ denotes conjugate transpose. So it follows that

$$\mathbf{x}(n)=\begin{bmatrix} h_1 & h_2 \\ h_2^* & -h_1^* \end{bmatrix}\bullet\begin{bmatrix} s_1(n-d_1) \\ s_2(n-d_2) \end{bmatrix}+\begin{bmatrix} v_1(n) \\ v_2^*(N-n+d_1+d_2) \end{bmatrix}\triangleq \mathbf{H}\bullet\mathbf{s}(n)+\mathbf{v}(n) \ . \tag{3}$$

The STBC coded transmission scheme achieves full diversity with full rate, and save optimal energy and bandwidth efficiency in this way.

2.2 Asynchronous STBC Decoding Scheme for Cooperative Transmission

Because of transmission delays, the traditional STBC decoding can not be applied. Transmission delays and channels can be estimated efficiently from training sequence. With similar decoding technology as in [6], we could obtain estimated d_1, d_2 and estimated $\mathbf{H}$ through the formula 6 and 8 respectively as in [6]. In the case, we could estimate symbol vector $\hat{\mathbf{s}}(n)$ by (4).

$$\hat{\mathbf{s}}(n) = (|h_1|^2 + |h_2|^2)^{-1} \bullet \mathbf{H}^H \bullet \mathbf{x}(n) \ . \tag{4}$$

The STBC encoding and decoding scheme of the cooperative transmission considering delay asynchronism without signal space diversity is described in detail in this section. And then we will present the new cooperative transmission considering signal space diversity in the next section.

3 The New Cooperative Transmission Scheme with Signal Space Diversity

Signal Space diversity, also known as modulation diversity, is proposed by Boutros [8]. Fig.2. shows the system model of rotated constellation with interleaving over coordinates. In this section, we will describe the new cooperative transmission scheme with signal space diversity.

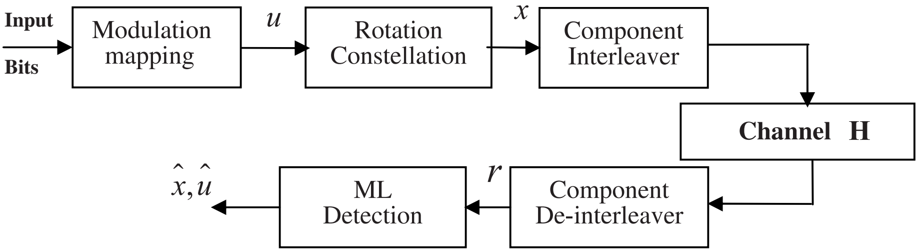


Fig. 2. System Model of rotated constellation with interleaving over coordinates

3.1 Signal Space Diversity with Coordinate Interleaving and Constellation Rotation

Signal Space diversity, mainly includes two components, which are coordinate interleaving and constellation rotation. And then, we introduce them respectively in the following.

A two-dimensional QPSK signal can be modeled as a combination of two orthogonal PAM signals: in-phase (I) and quadrature (Q) signals, transmitted over the same channel. This implies that both signal components can be fade independently if they are not concurrently transmitted during the same fading interval. As in [9, 10], we use the coordinate interleaving between I and Q components to provide independent fading. However, interleaving could lead to large delays and a large amount of memory when the channel slowly varying. We use the practical solution to replace coordinate interleaver with a time delay in only one of the quadrature components. We assume that the delay exceeds the average duration of a fade period.

Given a coordinate interleaved symbol mapped to a regular constellation, we would like to apply a transformation to the constellation which preserves the Euclidean distances between points, but improves the constellation's resistance to fading. Because that the transformation preserves Euclidean distances and norms, and the rotated signal set has the same performance of the non-rotated one when used over the AWGN channel, namely the performance of the AWGN channel is not affected by the rotated constellation transformation [9].

The diversity order can therefore be increased by making all distinct signal symbols exhibited distinguishable coordinates with independent fading on each component of the same signal symbol.

Here, we maximize the minimum squared product distance d_p^2 considering the secondary criterion but the optimal criterion which is to maximize the minimum Hamming distance. For the Rayleigh fading channel, minimum squared product distance d_p^2 [12], is named by (5)

$$d_p^2 = \min_{m,\widehat{m}\in C} \prod_{\substack{i=1 \\ m_i \neq \widehat{m}_i}}^{M} (m_i - \widehat{m_i})^2 \ . \tag{5}$$

Where m_i and $\widehat{m_i}$ denotes M-dimensional estimated and transmitted symbols respectively.

With the criterion of the minimum squared product distance, the real rotation matrix for M=2 is given by [12], and by varying only one rotation angle, optimal rotated constellation is formed. The optimal angle of rotation for this constellation can be found using maximized d_p^2 and it is $\phi_{opt} = 31.7^{\circ}$ [9, 12].

3.2 The New Transmission Schemes with Signal Space Diversity

The block diagram for cooperative transmission without perfect synchronization with a rotated constellation where the signal in a quadrature channel is delayed before being transmitted is shown in Fig.3. A modulator maps each channel symbol v to a complex signal $s(n) = \mu(v) = s_I(n) + js_Q(n)$ chosen from a rotated constellation set.

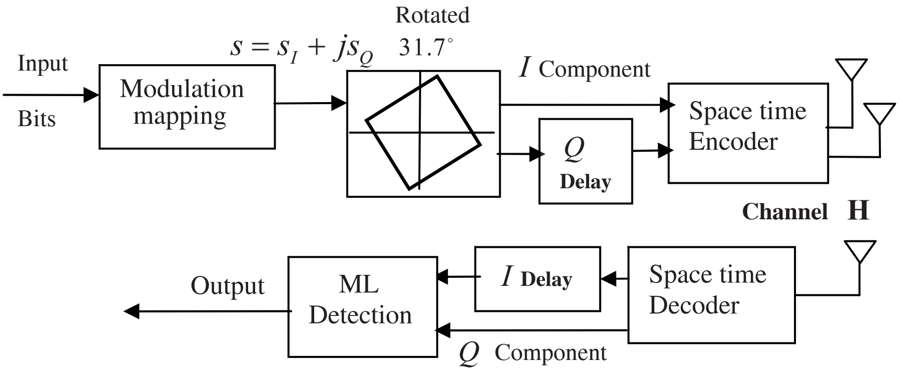


Fig. 3. New transmission scheme for wireless sensor networks with signal space diversity

With delaying k modulation symbols in the Q component in the transmitter, namely $s_{t,Q_k}(n)=\mu(v_t)=s_I(n)+js_{Q-k}(n)$, and at the receiver, we could de-interleave by delaying k symbols in the I component, namely $s_{t,I_k,Q_k}(n)=s_{I-k}(n)+js_{Q-k}(n)$.

In the symbol period 1, sensor1 transmits $\mathrm{S1}':\{s_{1,\mathrm{Q_k}}(1),\cdots,s_{1,\mathrm{Q_k}}(\mathrm{N})\}$ while sensor 2 transmits $\mathrm{S2}':\{s_{2,\mathrm{Q_k}}(1),\cdots,s_{2,\mathrm{Q_k}}(\mathrm{N})\}$ simultaneously to the hop $i+1$. During the next symbol period, the sensor 1 transmits $\{-s^*_{2,Q_k}(N),\cdots,-s^*_{2,Q_k}(1)\}$ and sensor 2 transmits $\{s^*_{1,Q_k}(N),\cdots,s^*_{1,Q_k}(1)\}$ to the hop $i+1$, where $(\bullet)^*$ denotes complex conjugate.

Similarly to section 2.1, we also consider delay asynchronism, and assume transmission delay of sensor 1 is d_1 and delay of sensor 2 is d_2. For fading channels, we consider nodes 1 and 2 transmit data packets to the next hop through Rayleigh flat-fading with coefficients h_1' and h_2' respectively and keep constant over the two symbol periods with changing randomly subsequently. The signals received at the receiver over the two consecutive periods are $x_1'(n)$ and $x_2'(n)$, respectively. Note N samples per symbol period, and $1\le n\le N$.

$$x_1'(n)=[h_1' \quad h_2']\bullet\begin{bmatrix} s_{1,Q_k}(n-d_1)\\ s_{2,Q_k}(n-d_2)\end{bmatrix}+v_1'(n)\ . \tag{6}$$

$$x_2'(n)=[h_1' \quad h_2']\bullet\begin{bmatrix} -s^*_{2,Q_k}(N-n+d_1)\\ s^*_{1,Q_k}(N-n+d_2)\end{bmatrix}+v_2'(n)\ . \tag{7}$$

Where $v_i'(n), i=1,2$ are $1\times N$ noise vectors whose elements are uncorrelated, and are zero mean complex Gaussian random variable with variance σ^2 . $s_{1,Q_k}(n)=s_{2,Q_k}(n)=0 \quad if\, n<1 \quad or \quad n>N$. Note sample vector $\mathbf{x}'(n)=\begin{bmatrix} x_1'^*(n)\\ x_2'(N-n+d_1+d_2)\end{bmatrix}^H$, where $(\bullet)^H$ denotes conjugate transpose. So it follows that

$$\begin{aligned}\mathbf{x}'(n)&=\begin{bmatrix} h_1' & h_2'\\ h_2'^* & -h_1'^*\end{bmatrix}\bullet\begin{bmatrix} s_{1,Q_k}(n-d_1)\\ s_{2,Q_k}(n-d_2)\end{bmatrix}+\begin{bmatrix} v_1'(n)\\ v_2'^*(N-n+d_1+d_2)\end{bmatrix}\\ &\triangleq \mathbf{H}'\bullet\mathbf{s}_{Q_k}(n)+\mathbf{v}'(n)\ .\end{aligned} \tag{8}$$

For simplification, we consider the receiver could estimate the cooperative transmission delay and channel state information depending on the pseudo-noise training sequences without coordinate interleaved. Because of limited space, we skip the estimated process for short.

With the estimated d_1 d_2 and $\mathbf{H}'$, the estimated symbol vectors $\hat{\mathbf{s}}_{Q_k}(n) = \hat{s}_I(n) + j\hat{s}_{Q-k}(n)$ can be obtained by (9). In order to obtain the channel fading coefficient for the I component of the signal, we calculate the formula (10) with only delaying k symbols for the I component.

$$\mathbf{H}'^{H} \bullet \mathbf{x}'(n) = (|h_1'|^2 + |h_2'|^2) \bullet \hat{\mathbf{s}}_{Q_k}(n) = (|h_1'|^2 + |h_2'|^2) \bullet (\hat{s}_I(n) + j\hat{s}_{Q-k}(n)) \ . \tag{9}$$

$$\mathbf{H}_k'^{H} \bullet \mathbf{x}_k'(n) = (|h_{1,k}'|^2 + |h_{2,k}'|^2) \bullet \hat{\mathbf{s}}_{I_k,Q}(n) = (|h_{1,k}'|^2 + |h_{2,k}'|^2) \bullet (\hat{s}_{I-k}(n) + j\hat{s}_Q(n)) \ . \tag{10}$$

By de-interleaving for the estimated symbol vectors, namely delaying k symbols in the I component in the receiver, and using (10), then we finally could have the estimated value $\widehat{s_k}(n) = \hat{s}_{I-k}(n) + j\hat{s}_{Q-k}(n)$ or $\hat{s}(n) = \hat{s}_I(n) + j\hat{s}_Q(n)$ for original transmitted symbols as (11).

$$\begin{aligned} \hat{u}(n) &= (|h_1'|^2 + |h_2'|^2) \bullet \hat{s}_I(n) + j(|h_{1,k}'|^2 + |h_{2,k}'|^2) \bullet \hat{s}_Q \\ \hat{u}_k(n) &= (|h_{1,k}'|^2 + |h_{2,k}'|^2) \bullet \hat{s}_{I-k}(n) + j(|h_1'|^2 + |h_2'|^2) \bullet \hat{s}_{Q-k}(n) \end{aligned} \tag{11}$$

Where $\hat{u}(n)$ and $\hat{u}_k(n)$ denote the input symbols to the maximum likelihood detector. The maximum likelihood (ML) detector that uses a decision criterion with estimated $\mathbf{H}'$ and $\mathbf{H}_k'$ is as follows as (12).

$$\begin{aligned} &\min\{|\mathrm{Re}[\hat{u}(n) - (|h_1'|^2 + |h_2'|^2)\hat{s}(n)]|^2 + |\mathrm{Im}[\hat{u}(n) - (|h_{1,k}'|^2 + |h_{2,k}'|^2)\hat{s}(n)]|^2\} \\ &\min\{|\mathrm{Re}[\hat{u}_k(n) - (|h_{1,k}'|^2 + |h_{2,k}'|^2)\hat{s}_k(n)]|^2 + |\mathrm{Im}[\hat{u}_k(n) - (|h_1'|^2 + |h_2'|^2)\hat{s}_k(n)]|^2\} \end{aligned} \tag{12}$$

The in-phase signal $s_I(n)$, and the quadrature signal $s_Q(n)$ of $s(n)$, experience channel h_1' , h_2' and $h_{1,k}'$, $h_{2,k}'$, respectively. These contribute to the diversity of the system. Due to the introduction of the constellation rotation, each of these components provides the complete signal of $s(n)$ [9]. By introducing the signal space diversity into the cooperative transmission, the new scheme can obtain lower SER and performance simulation is described in the next section in detail.

4 Performance Analysis and Numerical Results

In this section, we will compare the new transmission scheme using signal space diversity with the traditional transmission and cooperative transmission, and symbol-error rate (SER) and bit-error rate (BER) were used as criteria. Each data packet contained 2000 QPSK symbols, of which 150 were training. The channel is modeled as an independent Rayleigh fading channel, and keeps constant over the two symbol periods with varying randomly afterwards. We assume that the transmission delay is the integer d , and $d \leq D = 5$. We used Monte-Carlo runs to evaluate each SER or BER with respect to the average received SNR (signal to noise). The total transmission power is assumed to be same with the same SNR for single-transmission scheme and cooperative transmission scheme with or without signal space diversity.

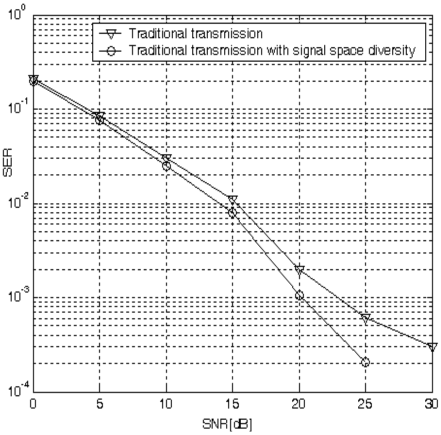


Fig. 4. Symbol-error-rate comparison between traditional transmissions with or without signal space diversity against received *SNR*

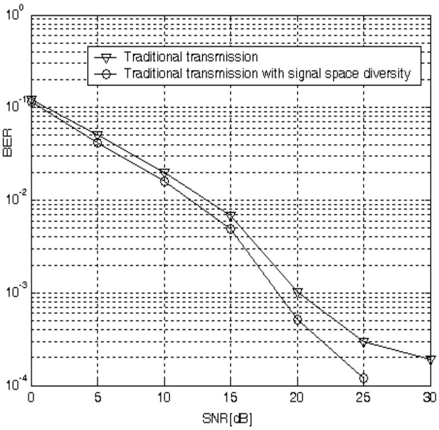


Fig. 5. Bit-error-rate comparison between traditional transmissions with or without signal space diversity against received *SNR*

Firstly, we will compare between traditional transmissions with or without signal space diversity as in Fig.4 and Fig.5. For traditional single-transmit single-receive transmission, synchronization is not a problem. From Fig.4 or Fig.5, we could observe that due to signal space diversity, the performance of the traditional transmission is improved. For example, in Fig.4, we could see, to obtain SER 0.1% , compared with the system with signal space diversity, the traditional transmission wastes almost $2.7dB$ SNR. Similar to Fig.4, Fig.5 also could be analyzed.

Fig.6 and Fig.7 respectively show the SER and BER comparison of asynchronous STBC-encoded cooperative transmission for wireless sensor networks and traditional transmission with or without signal space diversity.

As in Fig.6, when not considering signal space diversity, the asynchronous STBC-encoded cooperative transmission has lower SER than the traditional transmission with the same transmission power, and because transmission diversity is exploited to mitigate channel fading. For example, to achieve SER 1% , the cooperative

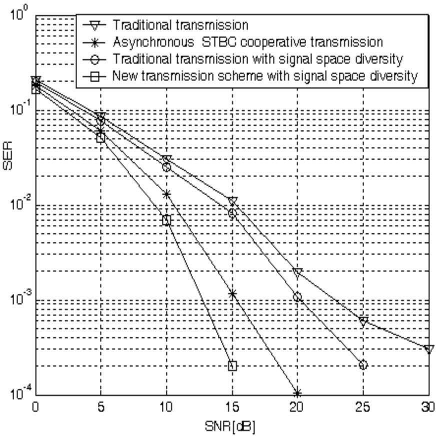


Fig. 6. Symbol-error-rate comparison between asynchronous STBC cooperative transmission and traditional transmission with or without signal space diversity against received *SNR*

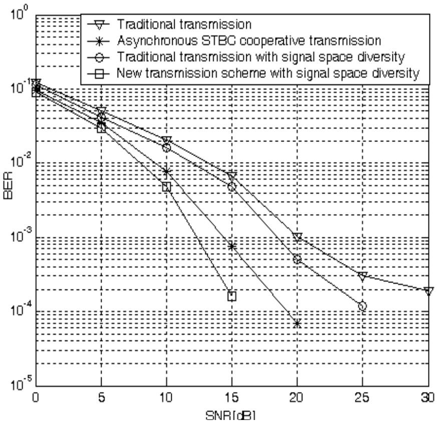


Fig.7. Bit-error-rate comparison between asynchronous STBC cooperative transmission and traditional transmission with or without signal space diversity against received *SNR*

transmission only requires $10dB$ SNR and $5dB$ SNR less than traditional transmission, thus saved energy and sensor lifetime could be prolonged by about three times. And to achieve SER 0.1% , the cooperative transmission could save about $8dB$ SNR in contrast to the traditional transmission and senor lifetime is certainly prolonged.

Considering signal space diversity, both asynchronous STBC-encoded cooperative transmission and traditional transmission have lower SER than that of not considering signal space diversity with the same total transmission. Because the signal space diversity increases diversity order through involving interleaving over coordinates and choosing a proper rotated signal constellation. Because of signal space diversity, to achieve SER 1% , the cooperative transmission without perfect synchronization only requires $9dB$ SNR, $1dB$ SNR less than original transmission scheme, $6dB$ SNR less than traditional transmission without signal space diversity. Furthermore, when obtaining SER 0.1% , the new scheme could save about $2.7dB$ SNR comparing to the cooperative transmission without signal space diversity, and save almost half SNR of which the traditional transmission requires, namely almost $12.5dB$. Similarly, from Fig.7, we could observe approximate change trend. Therefore, utilizing signal space diversity, both traditional transmission and cooperative transmission could have lower SER and BER, and system performance is enhanced.

5 Conclusion

In this paper, we have proposed the new scheme for asynchronous STBC-encoded cooperative transmission using signal space diversity. In order to increase the diversity advantage, we applied coordinate interleaving and the optimum constellation rotation into the asynchronous cooperative transmission. The diversity of the new transmission scheme is resulted from Space-Time transmission diversity and Signal-Space diversity. Simulation results have shown that the new transmission scheme could improve transmission diversity order and obtain better system performance.

References

1. Li, X., Chen, M., Liu, W.: Application of STBC-encoded cooperative transmissions in wireless sensor networks. IEEE Signal Processing Lett. (2005)
2. Li, X., Ng, F., Hwu, J., Chen, M.: Channel equalization for STBC-encoded cooperative transmissions with asynchronous transmitters. In: Proc. 39th Asilomar Conf. Signals, Systems Computers, Pacific Grove, CA (2005)
3. Alamouti, S.M.: A simple transmitter diversity scheme for wireless communications. IEEE J. Select. Areas Commun. 16, 1451–1458 (1998)
4. Akyildiz, I.F., et al.: A survey on sensor networks. IEEE Commun. Mag. 40(8), 102–114 (2002)
5. Li, X.: Space-time coded multi-transmission among distributed transmitters without perfect synchronization. IEEE Signal Process. Lett. 11(12), 948–951 (2004)
6. Li, X.: Energy efficient wireless sensor networks with transmission diversity. Electron. Lett. 39(24), 1753–1755 (2003)
7. Boulle, K., Belfiore, J.C.: Modulation scheme designed for the Rayleigh fading channel. In: Proc. Conf. Information and Systems, pp. 56–67. Princeton, NJ (1992)
8. Boutros, J., Viterbo, E.: Signal Space Diversity: a power and bandwidth efficient diversity technique for the Rayleigh fading channel. IEEE Transactions on information Theory 44(4), 1453–1476 (1998)
9. Kim, Y.-H., Kaveh, M.: Coordinate interleaved space time coding with rotated constellation. In: IEEE conference on Vehicular Technology, pp. 732–735 (2003)
10. Chindapol, A., Ritcey, J.A.: Bit-interleaved coded modulation with signal space diversity in Rayleigh fading. In: Proc. 33rd Asilomar Conf. Signals, Systems, Computers, Pacific Grove, CA, pp. 1003–1007 (1999)
11. Correia, A., Hottinen, A., Wichman, R.: Optimized Constellations for Transmitter Diversity. In: IEEE Vehicular Technology Conference, Fall, Amsterdam, Holland, the Netherlands, vol. 3, pp. 1785–1789 (1999)
12. DaSilva, V.M., Sousa, E.S.: Fading-resistant modulation using several transmitter antennas. IEEE Trans. Commun. 45, 1236–1244 (1997)
13. Chindapol, A., Ritcey, J.A.: Design, analysis, and performance evaluation for BICM-ID with square QAM constellations in Rayleigh fading channels. IEEE Journal on Selected Areas in Communications 19(5), 944–957 (2001)
14. Wang, H., Wang, D., Xia, X.-G.: On optimal quasi orthogonal space-time block codes with minimum decoding complexity. In: Proc. ISIT 2005, pp.1168–1172 (2005)

On the Temporal-Spatial Correlation Based Fault-Tolerant Dynamic Event Region Detection Scheme in Wireless Sensor Networks

Bo Yao and Qingchun Chen

Provincial Key Lab of Information Coding & Transmission
Southwest Jiaotong University, Chengdu, Sichuan 610031, China
jimmy_yao2002@163.com, qcchen@home.swjtu.edu.cn

Abstract. One of the important sensor network applications is monitoring inaccessible environments. The noisy environment and energy constraints, however, challenge the event detection problem. Most of recently proposed fault-tolerant event detection algorithms are only based on the spatial correlation. In these algorithms, the frequent exchanges of measurements among nearby sensor nodes give rise to much energy dissipation. Moreover, detection accuracy is poor at the boundary of event region and the edge of sensor networks. In this paper, we propose a temporal-spatial correlation based fault-tolerant event region detection algorithm. In order to improve the performance, both spatial correlated information and temporal correlated information are employed in the event detection. It is validated through simulations that, the proposed temporal-spatial correlation based algorithm outperforms the spatial correlation based scheme in terms of detection accuracy and energy dissipation, thus making the proposed algorithm attractive in energy-efficient event region detection applications.

Keywords: Threshold zone, fault tolerance, event detection, temporal-spatial correlation, wireless sensor networks.

1 Introduction

Recent advances in wireless communications and the micro-electro-mechanical systems (MEMS) have enabled the development of low-cost wireless senor networks. Although each sensor node has limited computation, communication, and sensing capability, hundreds or thousands of sensor nodes could be densely deployed to enable powerful and robust environment monitoring in an unattended mode. These features make the wireless sensor networks attractive in a wide range of applications like the industrial automation, military tactical surveillance, national security, and emergency health care [1]~[3]. Among all applications, the event region detection problem was investigated extensively in recent years. The event region detection is referred to the particular class of queries of determining the location of event regions in the environment with some distinguishable characteristics, such as an unusual chemical concentration and hazardous radiation that generates safety and health concerns for the public [2], [3]. However, the limited processing capability of individual sensor node and the constrained energy resources impose challenges on the

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 511–523, 2007.

event detection problem in terms of detection accuracy and energy-efficiency. Thus, energy-efficient and fault-tolerant event detection algorithms are required for wireless sensor network applications.

By examining the spatial correlation in the readings of nearby sensors, a distributed probabilistic Bayesian fault-tolerant algorithm was proposed in [2] to disambiguate and correct the faulty sensor readings. The distributed fault-tolerant algorithm is based on the fact that, the sensor faults are likely to be stochastically independent, while the event measurements are likely to be spatial correlated due to the densely deployed assumption. Furthermore, a majority voting scheme is shown to be optimal in terms of the fault correction capability [2], [3]. The distributed event detection scheme is simple to be realized since only local information dissemination is needed. However, there may be much energy dissipation caused by the frequent exchanges of measurements among nearby sensors in event detection applications. In addition, detection accuracy is poor at the boundary of event region and the edge of whole sensor networks, where there is conflict in sensor readings along the boundary of some event region and less correlated (redundant) information (less neighboring nodes) available at the edge of network.

A moving-median based distributed algorithm was proposed in [8], wherein float format sensor readings are exchanged among nearby sensor nodes. As expected, the detection accuracy will be improved significantly by employing the soft-valued information during the event detection. However, this algorithm is not energy-efficient since the communication cost will be much higher than that of the hard-decision majority voting algorithm in [2]. In order to minimize the communication overhead during the event detection procedure, a search algorithm was proposed in [4] to determine the minimum number of neighboring nodes to achieve the specified detection accuracy. To balance the detection accuracy and the energy dissipation, the efficient noise-tolerant event detection (NED) algorithm was developed in [5], wherein a variable length coding mechanism is used to represent partial estimation results exchanged locally among neighboring nodes. The communication cost of NED, nevertheless, is still higher than that of the majority voting scheme in [2].

In this paper, an energy-efficient fault-tolerant dynamic event detection algorithm is proposed by employing both the temporal and spatial correlation property in the wireless sensor networks. Here the temporal correlation implies that, the physical characteristics in the environment generally fluctuate slowly. As an example, the chemical concentration in the air will remain unchanged for comparative long time duration. Thus, faults due to the environment noise and surrounding interference could be corrected to some extent by examining the temporal correlations from the successive measurements by the same well-behaved sensor node. For this reason, neighboring measurement retrievals could be saved to achieve the reduction in the energy dissipation, as computation is generally much cheaper than communication [6], [7]. However, if the characteristic in the environment does change, the spatial correlation information could also be exploited to respond quickly to the variations. It is shown through simulation results that, the temporal-spatial correlation based fault-tolerant distributed event detection scheme can achieve better detection accuracy, while dissipating less energy than the spatial correlation based scheme.

The remainder of the paper is organized as follows. In section 2, a brief survey about the distributed event detection scheme will be presented. After a comparative discussion of both the spatial correlation concept and the temporal correlation concept, the spatial-temporal correlation based fault tolerant event detection scheme is presented in section 3. Simulation results are presented in section 4 to validate the applicability of the proposed scheme. Finally, a summary is presented in section 5 to conclude this paper.

2 Event Region Detection

In the event region detection application, sensor nodes are placed in an operational environment, and tasked to identify the regions in the network that contain interesting events in a self-organized manner [2]. It is generally assumed that each sensor knows its own geographical location, either through GPS or through RF-based beacons. As an example scenario in Fig. 1, we have a grid of sensors in some operational area. There is an event region with unusually high chemical concentration. Some of the sensors may be faulty and report erroneous readings due to the noise and interference within the environment.

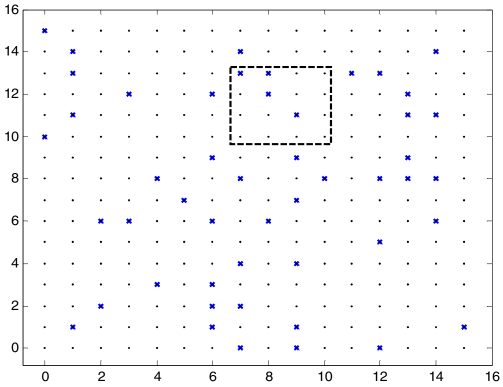


Fig. 1. Sample scenario: A distributed sensor network with uncorrelated sensor faults (denoted as "×") deployed in an environment with a single event (enclosed inside the square with the bold dashed line)

The first step in event region detection is to determine which sensor readings are interesting. Without loss of generality, it could be assumed that, a user-specified threshold value of *th* that enables each node to determine whether its reading corresponds to an event or not has been specified with the query or otherwise made available to the nodes during deployment [2]. Let $x_i(t)$ denote the measurement of sensor node *i* at time *t*:

$$x_i(t) = m_i(t) + z_i(t), \tag{1}$$

where $m_i(t)$ represents the value of characteristic in detected environment at the *i*-th node at time *t*, while $z_i(t)$ indicates the associated noise and interference. Without loss of generality, it is assumed that $z_i(t)$ is modeled as a Gaussian random variable with mean 0 and standard deviation σ. Each sensor will make up a decision by comparing its periodic measurement reading of $x_i(kT_s)$ with the threshold value of th, where T_s is the detection interval that should be set according to the nature of the monitored object and the energy-efficient requirement by system[1]. Let the real situation of the i-th sensor node at time kT_s be modeled by a binary variable $T_i(kT_s)$, while the real output of sensor be mapped into a binary variable $S_i(kT_s)$. Then we have

$$T_i(kT_s) = \begin{cases} 1, & if \quad m_i(kT_s) \geq th \\ 0, & if \quad m_i(kT_s) < th \end{cases}, \quad S_i(kT_s) = \begin{cases} 1, & if \quad x_i(kT_s) \geq th \\ 0, & if \quad x_i(kT_s) < th \end{cases}. \tag{2}$$

That is, $T_i(kT_s)$=1 means that there is an event and $S_i(kT_s)$=1 implies that there is a reported event; on the other hand, $T_i(kT_s)$=0 means that there is no event and $S_i(kT_s)$=0 implies that there is no reported event. And the following two probabilities correspond to the faulty readings

$$\begin{aligned} p_1 &= P(S_i(kT_s) = 0 \mid T_i(kT_s) = 1) = P(x_i(kT_s) < th \mid m_i(kT_s) \geq th) = Q\left(\frac{m_i(kT_s) - th}{\sigma}\right) \\ p_2 &= P(S_i(kT_s) = 1 \mid T_i(kT_s) = 0) = P(x_i(kT_s) \geq th \mid m_i(kT_s) < th) = Q\left(\frac{th - m_i(kT_s)}{\sigma}\right), \end{aligned} \tag{3}$$

where the $Q(x)$ function is defined as below

$$Q(x) = \int_x^{\infty} \frac{1}{\sqrt{2\pi}} e^{-y^2/2} dy. \tag{4}$$

It must be addressed that, symmetric error probability is generally assumed, i.e., p_1=p_2 [2]. But from (3) we can conclude that, for a given threshold value of th, both p_1 and p_2 are dependent on both reading of $m_i(kT_s)$ and the standard deviation σ. In the previous spatial correlation event detection scheme, only the current measurement of $m_i(kT_s)$ is used. In this paper, successive measurements are also considered to be included in the dynamic event region detection. Our objective is to develop an effective algorithm to make full use of both the spatial and temporal correlation to enable energy-efficient dynamic event region detection mechanism.

3 The Temporal-Spatial Correlation Based Dynamic Event Region Detection Scheme

In this section, we present both the temporal correlation based dynamic event region detection scheme and the temporal-spatial correlation based dynamic event region detection scheme. Before we step into the fault tolerant dynamic event region detection scheme, the spatial correlation concept will be reviewed at first.

Basically, it is a challenging task to disambiguate events from faults in the sensor readings since an unusually high reading could potentially correspond to both event

[1] Due to the slow environment variation property and the energy efficient requirement, each sensor just needs to read its measurement result periodically.

and normal situation. Conversely, a faulty node may report a low measurement even though it is in an event region. Without loss of generality, the event measurements among neighboring nodes are spatially correlated when sensors are densely deployed. As a result, nearby sensors are likely to have similar measurements. This kind of spatial redundancy information is referred to as the *spatial correlation* in this paper. In [2], probabilistic decoding mechanisms are presented by employing the spatial correlations. The idea is that, sensor faults are likely to be stochastically uncorrelated, while nearby sensors are likely to have the similar event readings unless they are at the boundary of the event region. Thus, faults could be identified and corrected by examining the correlation in the readings of nearby sensors.

3.1 Temporal Correlation Based Dynamic Event Region Detection Scheme

In case of the characteristic of the phenomenon detected by the sensor networks fluctuates slowly, for example, the chemical concentration in the air or water and the temperature in an indoor environment. That is to say, the interested signal $m_i(t)$ can be considered as constant in a short time duration, i.e., in most time, there is no event in the detected environment, and if event happens, it will last for a while. We can exploit this to disambiguate the event from noise-related measurement error, which is referred to as *temporal correlation* in this paper.

If the characteristic in the environment detected by node i remains unchanged in time duration C=[(k-L)T_s, kT_s], there are in total (L+1) consecutive redundant measurements that could be utilized to determine whether there is an event or not. Let $\overline{X}_{i;C}$ be the mean measurement of node i in time duration C, and we have

$$\overline{X}_{i;C} = \frac{1}{L+1}\sum_{l=0}^{L} x_i\left[(k-l)T_s\right] = \frac{1}{L+1}\sum_{l=0}^{L} m_i\left[(k-l)T_s\right] + \frac{1}{L+1}\sum_{l=0}^{L} z_i\left[(k-l)T_s\right] = \overline{m}_i + \overline{z}_i, \tag{5}$$

where $\overline{m_i}$ denotes the real average measurement in time duration C, and $\overline{z}_i$ represents a Gaussian random variable with mean 0 and variance $\sigma^2/(L+1)$. In order to simplify the analysis, we follow the same assumption as that in [2], that is, let m_n be the mean normal measurement and m_f be the mean event measurement, then $\overline{m_i}$ could be either m_n or m_f, and the binary decision threshold th=0.5(m_n+m_f). With loss of generality, it is assumed that m_n<m_f in this paper. Now, we can determine a binary estimate $R_i(kT_s)$ for the true binary reading $T_i(kT_s)$ at time kT_s as below

$$R_i(kT_s) = \begin{cases} 0 & if \quad \overline{X}_{i;(k-L)T_s,kT_s} < th \\ 1 & if \quad \overline{X}_{i;(k-L)T_s,kT_s} \geq th. \end{cases} \tag{6}$$

And the error probability in event detection can be evaluated approximately by using the following Q-function

$$\begin{aligned} \hat{p} &= P\big(R_i(kT_s)=0 \mid T_i(kT_s)=1\big) = P(R_i(kT_s)=1 \mid T_i(kT_s)=0) \\ &= Q\Big[(m_f - m_n)\sqrt{L+1}/2\sigma\Big] \end{aligned} \tag{7}$$

In particular, $P_M = P(R_i(kT_s) = 0 \mid T_i(kT_s) = 1)$ denotes the missing alarm probability and $P_F = P(R_i(kT_s) = 1 \mid T_i(kT_s) = 0$ denotes the false alarm probability. The missing alarm probability is not equal to the false alarm probability if the predefined threshold *th* is not equal to $0.5(m_n + m_f)$. And $S_i(kT_s) \neq T_i(kT_s)$ *and* $R_i(kT_s) = T_i(kT_s)$ implies a detection error correction. It can be observed from (7) that the estimate error probability decreases as *L* increases. In other words, the longer the characteristic of phenomenon remains unchanged, the lower the estimate error probability will be.

The temporal correlation-based scheme is based on the assumption that, in most time, the characteristic of phenomenon will remain unchanged for a while. However, how can we decide whether the characteristics in the detected environment change or not? In order to solve this problem, we propose the *threshold zone* concept as follows.

> ***Definition:*** *The threshold zone is defined as* $[m_f - a\sigma,\ m_n + a\sigma]$*, where a is the parameter corresponding to the threshold zone range.*

For example, the following two scenarios could be regarded as an *obvious* change in the detected environment.

$$\begin{aligned} &\{x_i((k-1)T_s) < m_f - a\sigma \quad and \quad x_i(kT_s) > m_n + a\sigma\} \\ &\{x_i((k-1)T_s) > m_n + a\sigma \quad and \quad x_i(kT_s) < m_f - a\sigma\}. \end{aligned} \tag{8}$$

Otherwise, we may assume that there is no obvious change in the characteristic to be detected. If a measurement $x_i(kT_s)$ changes obviously compared with $x_i((k\text{-}1)T_s)$, the characteristic may change from the normal to event, or event to the normal, with different certainty at different *a*. For a given mean normal measurement m_n and the mean event measurement m_f, the measurement of $x_i(kT_s)$ could be approximately assumed to be within either $[m_f - a\sigma,\ m_f + a\sigma]$ or $[m_n - a\sigma,\ m_n + a\sigma]$ if *a* is large enough[2]. Seemingly, the event detection may be ambiguous when the measurement of $x_i(kT_s)$ is within the range of $[m_f - a\sigma,\ m_n + a\sigma]$. Because *a* should be chosen to satisfy the inequality of $m_f - a\sigma < th$ *and* $m_n + a\sigma > th$, we get

$$a \geq \max\{(m_f - th)/\sigma, (th - m_n)/\sigma\}\,. \tag{9}$$

The basic idea of the threshold zone concept is to extend the conventional single threshold value to a threshold range. By doing so, one can determine whether the environment characteristic changes or not, and adjust the average observed measurements as in (5). If there is obvious change in the characteristic, the previous measurements will not be utilized to correct the present measurement; otherwise, the previous measurements will be examined to disambiguate faults in present measurement.

Based on the *threshold zone* concept, if there is no obvious change in the detected characteristic, the redundancy of the successive measurements could be utilized to suppress the noise effect, and therefore improve the detection accuracy by employing the following temporal correlation based ***Algorithm* 1**, wherein the *sliding window scheme* is employed to adapt to the environment variations.

[2] The probability that $x_i(kT_s)$ is either within $[m_f - a\sigma, m_f + a\sigma]$ or $[m_n - a\sigma, m_n + a\sigma]$ is $1 - 2Q(a)$.

Algorithm 1. Temporal Correlation-based Dynamic Event Region Detection Scheme

```
1: $L$: the length of sliding window, $w_i[L]$: the observed window of node $i$ to preserve data,
   $T_s$: the detection interval, $th$: binary decision threshold, $a$: threshold zone parameter.
2: $x_i(kT_s) \longleftarrow$ the measurement of node $i$ at time $kT_s$
3: if $k==1$ then
       $R_i(kT_s)= S_i(kT_s)$; $w_i[0]= x_i(kT_s)$; $j=0$;
   else if $x_i(kT_s)$ changes obviously compared with $x_i[(k-1)T_s]$ then
           $R_i(kT_s)= S_i(kT_s)$; $w_i[0]= x_i(kT_s)$; $j=0$;
       else $j$++;
   end if
4: if $1\leq j\leq L-1$ then
       $w_i[j]= x_i(kT_s)$; $mean=\frac{1}{j+1}\sum_{n=0}^{j} w[n]$;
       if $mean\geq th$ then $R_i(kT_s)=1$;
       else $R_i(kT_s)=0$;
       end if
   else if $j\geq L$ then
           for $n=0$ to $(L-2)$ do
               $w_i[n]= w_i[n+1]$;
           end for
           $w_i[L-1]= x_i(kT_s)$ ;
           $mean=\frac{1}{L}\sum_{n=0}^{L-1} w[n]$;
           if $mean\geq th$ then $R_i(kT_s)=1$;
           else $R_i(kT_s)=0$;
           end if
   end if
```

Compared with the optimal threshold decision scheme proposed in [2], the temporal correlation based detection algorithm does not need measurement exchange among nearby sensor nodes. Hence, no neighboring information exchange is required during the event detection and the energy consumption will be much lower. However, if obvious changes occur frequently, there is less redundancy of temporal correlated information can be used. As a result, missing alarm probability and false alarm probability with temporal based scheme is higher than that of majority voting algorithm, as will be illustrated in section 4.

3.2 Temporal-Spatial Correlation-Based Dynamic Event Region Detection Scheme

As discussed in the previous section, for the majority voting algorithm and other algorithms based on spatial correlation, frequent exchanges of measurements among neighboring nodes will cause high energy dissipation. The temporal correlation based algorithm consumes less energy, whereas, both the missing alarm probability and the false alarm probability increase when the characteristic changes quickly and frequently. To balance the energy dissipation and the detection accuracy requirement, we propose an energy-efficient fault-tolerant dynamic event detection scheme, wherein the temporal correlation is used in conjunction with the spatial correlation.

The basic idea of temporal-spatial fault-tolerant event detection scheme is that, when there has been no obvious change in the detected environment for a time duration C (i.e. C=LT_s) or longer than C, we exploit the temporal correlated information to disambiguate event from noise-related measurement error; when the characteristic changes obviously, we exploit the spatial correlated information to improve the event detection; when there has been no obvious change in the detected environment for a time duration shorter than C, we exploit both the temporal correlated information and spatial correlated information to disambiguate event from noise-related measurement error. And the proposed temporal-spatial correlation based ***Algorithm* 2** is presented in the following table.

Algorithm 2. Temporal-Spatial Correlation-based Dynamic Event Region Detection Scheme

```
1: $L$: the length of sliding window, $w_i[L]$: the observed window of node $i$ to preserve data,
$T_s$: the detection interval, $th$: binary decision threshold, $a$: threshold zone parameter, $\hat{S}_i(k.T_s)$:
the first estimate by node $i$ itself , $N_i$: number of neighboring nodes of node $i$.
2: $x_i(kT_s) \leftarrow$ the measurement of node $i$ at time $kT_s$
3: if $k$= =1 then
      $w_i[0]= x_i(kT_s)$; $j$=0;
      if $w_i[0] \geq th$ then
         $\hat{S}_i(T_s)=1$;
      else $\hat{S}_i(T_s)=0$;
      end if
   else if $x_i(kT_s)$ changes obviously compared with $x_i((k-1)T_s)$ based on threshold zone then
         $w_i[0]= x_i(kT_s)$; $j$=0;
      else $j$++;
   end if
4: if $1 \leq j \leq L-1$ then
      $w_i[j]= x_i(kT_s)$; $mean=\frac{1}{j+1}\sum_{n=0}^{j} w_i[n]$;
      if $mean \geq th$ then
         $\hat{S}_i(kT_s)=1$;
      else $\hat{S}_i(kT_s)=0$;
      end if
   else if $j \geq L$ then
      for $n$=0 to ($L$-2) do
         $w_i[n]= w_i[n+1]$;
      end for
      $w_i[L-1]= x_i(kT_s)$ ;
      $mean=\frac{1}{L}\sum_{n=0}^{L-1} w_i[n]$;
       if $mean \geq th$ then
          $R_i(kT_s)=\hat{S}_i(kT_s)=1$;
       else $R_i(kT_s)=\hat{S}_i(kT_s)=0$;
       end if
       go to 2
    end if
5: Obtain the first estimate $\hat{S}_m(k.T_s)$ of all $N_i$ neighbors of node $i$ , determine the final estimate
$R_i(kT_s)$ based on Optimal Threshold Decision Scheme in [2]
```

In particular, if there has been no obvious change in characteristic for a long time duration, few exchanges of neighboring information are needed, and hence, the effect of temporal-spatial correlation based dynamic event region detection scheme is nearly equivalent to that of temporal correlation based scheme. Because the characteristic of the detected environment is likely to change slowly in most applications, in such a case, the temporal-spatial correlation based scheme will be energy-efficient, as less exchanges of neighboring information are needed. On the other hand, if the characteristic changes obviously and frequently, frequent exchanges of neighboring information are needed, and the effect of temporal-spatial correlation based dynamic event region detection scheme is almost equivalent to that of the spatial correlation based scheme in [2]. And even though the characteristic changes frequently, the energy dissipated during detection of temporal-spatial correlation based scheme is less than that of the pure spatial correlation based scheme, as will be validated by the simulation results in section 4. It must be aware that, as for the gradual start event detection, detection delay will be introduced by using the proposed temporal-spatial correlation scheme due to the temporal averaging in the sliding window.

4 Simulation Results

In this section, we evaluate the performance of the temporal-spatial based dynamic event region detection algorithm presented in section 3. The results of the conventional spatial correlation based algorithm are presented for comparison purpose. All simulations are conducted in OPNET simulation platform. As for the spatial correlation based algorithm wherein the majority voting scheme is employed [2], we assume that the ground truth is the same for all neighbors of node i (i.e., we neglect the edge effects). And in simulations, we use the following parameters: the mean normal measurement m_n=0, the mean event measurement m_f=1, the threshold th=0.5, and the detection interval T_s=30s. In our simulations, a fixed 5% event probability is assumed, i.e., $P(T_i(kT_s)=1)$=5%, a fixed number of the neighboring sensor nodes N and a fixed sliding window size L are assumed to be 4 and 5, respectively, i.e., N= 4, and L=5.

The temporal correlation based detection algorithm with different time duration of event C is simulated to show the missing alarm probabilities P_M in Fig. 2, where the missing alarm probability of the conventional spatial correlation majority voting scheme is included as well for comparison. It is observed that, the missing alarm probability P_M decreases when the time duration of event increases. It complies with the previous discussion that, the temporal correlation based algorithm has better performance when the characteristic changes slowly. But the performance become awful when events change quickly, say, the time duration of event less than length of sliding window L. In addition, the performance differs for different parameter a of the threshold zone [1-$a\sigma$, $a\sigma$]. When the time duration of event is short, small parameter a is preferred, while larger a is preferred when event lasts a long duration. This fact could be intuitively explained as follows: Shorter time duration of event implies that a frequent event variation presents. In such case, less temporal correlation should be utilized by setting a smaller a value; otherwise, more temporal correlation

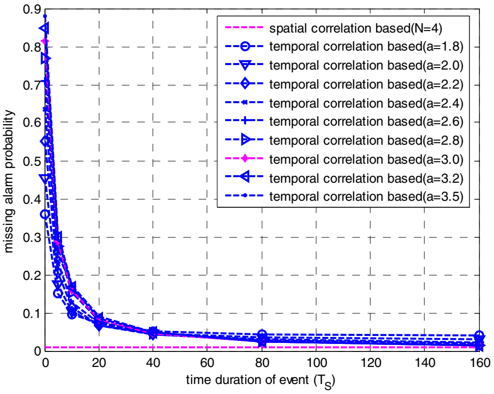
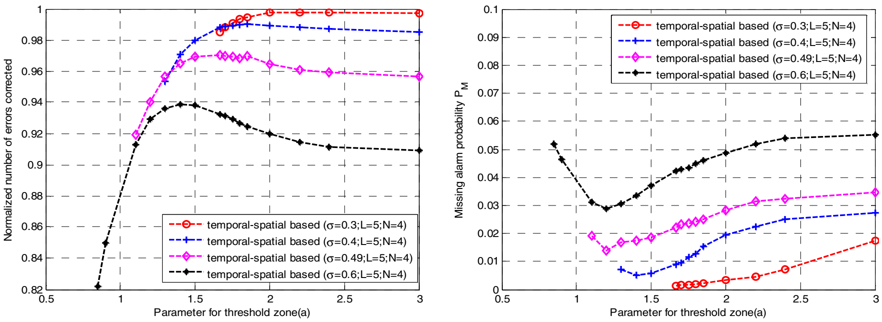


Fig. 2. The missing alarm probability P_M, temporal-correlation based scheme, $\sigma = 0.4$

should be used to achieve a better performance by setting a larger *a* value. Compared with that by the spatial correlation based scheme, nevertheless, higher missing alarm probability P_M will be achieved by employing the pure temporal correlation based algorithm.

The normalized number of error correction performance and the missing alarm probability of the temporal-spatial correlation based event detection algorithm are presented in Fig. 3(a) and Fig. 3(b), respectively. In both cases, fixed time duration of event is assumed, i.e., C=80T_s. It could be noted from Fig. 3(a) that, the temporal-spatial based scheme can correct up to 90% errors in event detection even when the standard deviation σ is as high as 0.6 (which corresponds to a sensor fault probability about 20%). For fixed noise standard deviation σ, an optimal parameter *a* of threshold zone [1-$a\sigma$, $a\sigma$] exists in terms of the maximum number of error corrections. It could also be observed that, there is also an optimal threshold range of [1-$a\sigma$, $a\sigma$]. It seems that, larger $a\sigma$ is preferable for larger σ. This fact could be intuitively explained in the following way. A larger σ implies that a high probability of variations in the measurements is owing to the noise instead of the change in the environment characteristic. Fig. 3(b) shows that, the missing alarm probabilities P_M's of the temporal-spatial correlation based scheme are very small, and larger P_M will be obtained for larger standard deviation. In addition, for the fixed noise standard deviation σ, an optimal parameter *a* of threshold zone [1-$a\sigma$, $a\sigma$] exists as well in terms of the minimized missing alarm probability P_M. It could be observed from the simulation results that, larger $a\sigma$ is also preferable with the increase in σ.

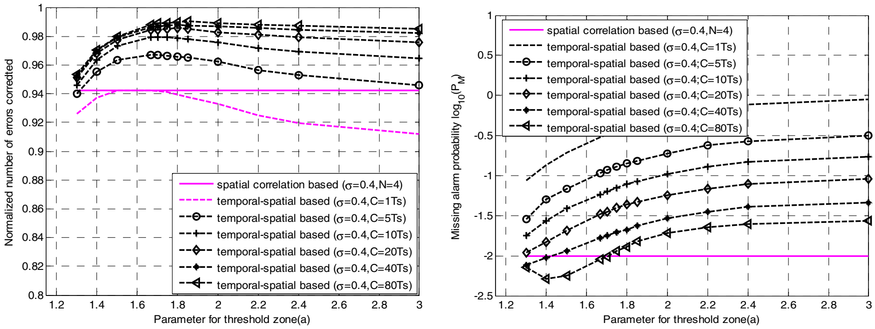
In Fig. 4(a) and Fig. 4(b), the pure spatial correlation based scheme and the proposed temporal-spatial correlation based scheme are compared in terms of the normalized number of error correction and the missing alarm probability. Unlike the previous simulations, different time durations of event *C* are simulated. It can be seen from Fig. 4(a) that, more errors will be corrected with longer time duration of event *C*. In addition, the temporal-spatial correlation based scheme generally outperforms the spatial correlation based scheme in terms of maximized error corrections. It can be



(a) Normalized number of errors corrected

(b) Missing alarm probability

Fig. 3. The error correction capability and the missing alarm probability, temporal-spatial correlation based scheme



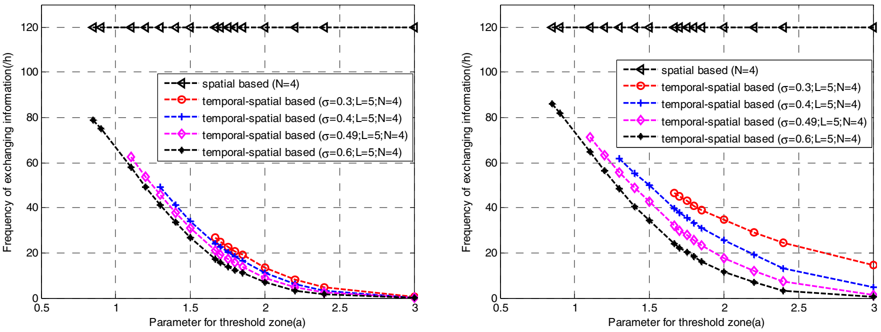
(a) Normalized number of errors corrected

(b) Missing alarm probability

Fig. 4. The error correction capability and the missing alarm probability, temporal-spatial correlation based scheme versus spatial correlation based scheme

seen from Fig. 4(b) that, there is larger missing alarm probability in the temporal-spatial correlation based scheme than in the spatial correlation based scheme, especially for the shorter event time duration, in that the temporal averaging in the temporal correlation is not suitable for the one-shot event measurement in its nature. However, it must be addressed that, there is few ‘one-shot’ event in the real practical applications. And less missing alarm probability can be achieved for the temporal-spatial based scheme than that of the spatial based scheme when C is large enough and a is small enough (for example, when $C \geq 40T_s$ and $a \leq 1.4$).

Since the energy consumption caused by computing is much cheaper than communicating [6], [7], we just take the energy dissipated by communicating into account in our simulation. For simplicity, information exchanging frequency is utilized to describe the energy consumption during the event detection procedure. In this paper, one step of information exchange includes both transmitting requirement information to neighbors and receiving data information from neighbors. In Fig. 5, the



(a) Time duration of event C=80T_s (b) Time duration of event C=1T_s

Fig. 5. The information exchanging frequency, temporal-spatial correlation based scheme versus spatial correlation based scheme

information exchanging frequency among neighboring nodes during the event detection is illustrated for different schemes, wherein fixed time duration of event C is assumed, i.e., C=80T_s in Fig. 5(a) and C=1T_s in Fig. 5(b). It can be seen that, much less information exchanges among neighboring nodes are needed in temporal-spatial correlation based scheme than that of the spatial correlation based scheme (majority voting in [2]). It can also be observed that, more information exchanges among neighbors are needed in temporal-spatial correlation based scheme when time duration of event C is shorter, but still less than that of the spatial correlation based scheme. Thus we conclude that, the proposed temporal-spatial correlation based scheme is much more energy-efficient than the spatial correlation based scheme.

5 Conclusion

In this paper, we focus on the fault-tolerant event detection problem in wireless sensor networks. The noisy environment and energy constraints, however, impose challenges on the event detection problem in terms of detection accuracy and energy-efficiency. Most of recently proposed fault-tolerant event detection algorithms are only based on the spatial correlation. In these algorithms, frequent exchanges of measurements among nearby sensor nodes give rise to much energy dissipation. Moreover, the detection accuracy is poor at the boundary of event region due to the measurement conflicts, and at the edge of sensor networks due to insufficient spatial redundancy. To achieve better detection accuracy and the energy consumption reduction, we propose temporal and temporal-spatial correlation based fault-tolerant event detection algorithms. The temporal correlation based scheme is energy-efficient, nevertheless, the missing alarm probability of which is higher than that of the spatial correlation based scheme. And the proposed temporal-spatial correlation based algorithm outperforms the spatial correlation based scheme in terms of detection accuracy and energy dissipation, especially when the detected environment changes slowly. It is also revealed through simulation results that, as for the proposed temporal-spatial correlation based scheme, there exist an optimal threshold parameter of a and an optimal threshold range [m_f–$a\sigma$, m_n+$a\sigma$] to achieve the best error correcting performance and the minimum missing alarm probability.

Acknowledgments

This work was supported by the NSFC under grant No. 60602044, the Fok Ying Tung Education Foundation under grant No.104026, and the Program for New Century Excellent Talents in University (NCET) under grant No.06-0797.

References

1. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: A Survey on Sensor Networks. IEEE Commun. Magazine 40(8), 103–114 (2002)
2. Krishnamachari, B., Iyengar, S.: Distributed Bayesian Algorithms for Fault-Tolerant Event Region Detection in Wireless Sensor Networks. IEEE Trans. Computers 53(3), 241–250 (2004)
3. Chen, Q.C., Lam, K.Y., Fan, P.Z.: Comments on Distributed Bayesian Algorithms for Fault-Tolerant Event Region Detection in Wireless Sensor Networks. IEEE Trans. Computers 54(8), 1182–1183 (2005)
4. Luo, X.W., Dong, M., Huang, Y.: On Distributed Fault-Tolerant Detection in Wireless Sensor Networks. IEEE Trans. Computers 55(1), 58–70 (2006)
5. Jin, G., Nittel, S.: NED: An Efficient Noise-Tolerant Event and Event Boundary Detection Algorithm in Wireless Sensor Networks. In: MDM 2006. Proc. of the 7th International Conference on Mobile Management, pp. 153–160. IEEE Computer Society, Nara (2006)
6. Heinzelman, W., Chandrakasan, A., Balakrishnan, H.: An Application-Specific Protocol Architecture for Wireless Microsensor Networks. IEEE Trans. Commun. 1(4), 660–670 (2002)
7. Heinzelman, W., Chandrakasan, A., Balakrishnan, H.: Energy-Efficient Communication Protocol for Wireless Microsensor Networks. In: Proc. of the 33rd Hawaii International Conference on System Sciences, pp. 3005–3014. IEEE Press, San Francisco (2000)
8. Ding, M., Chen, D., Xing, K., Cheng, X.: Localized Fault-Tolerant Event Boundary Detection in Sensor Networks. In: IEEE INFOCOM 2005, Miami, pp. 902–913 (2005)
9. D'Costa, A., Ramachandran, V., Sayeed, A.: Distributed Classification of Gaussian Space-Time Sources in Wireless Sensor Networks. IEEE J. Selected Areas in Commun. 22(6), 1026–1036 (2004)

Dynamic Simulation Based Localization for Mobile Sensor Networks*

Changming Su[1], Jiangwen Wan[2], and Ning Yu[1]

[1] Automation School, Beijing University of Posts and Telecommunications, Beijing, China
[2] School of Instrument Science and Optoelectronics Engineering, Beijing University of Aeronautics and Astronautics, Beijing, China
{su.changming, mail.yuning}@gmail.com
sensory@buaa.edu.cn

Abstract. In mobile wireless sensor networks, sensors can move randomly or keep static temporarily. Mobility makes the sensor networks better acquire information, but also makes accurate localization more difficult since the network environment changes continually. In this paper, an energy-efficient dynamic simulation based localization (DSL) algorithm is introduced that uses range measurement information to restrict sample region and establishes a dynamic filtering mechanism to improve the localization performance and efficiency. Analytical and simulation results are provided to study the localization cost and location accuracy in different mobility models and various environmental settings. The results indicate that our algorithm outperforms the best known simulation based localization schemes under a wide range of conditions, with localization accuracy improved by an average of 24% and computation cost reduced significantly for a similar high localization accuracy.

Keywords: Mobile Sensor Networks, Simulation Based Localization, Compuation cost.

1 Introduction

In mobile wireless sensor networks, a collection of mobile sensors are distributed over the monitored area. They can move randomly or keep static temporarily. The mobility of sensors makes the sensor network better acquire information. Therefor, it is important to have the knowledge of the location indicating where the data came from. But local network environment changes continually because of node's mobility. It makes the localization for mobile nodes much more difficult since the sensor is equipped with limited hardware.

Recently some localization techniques have been proposed to allow mobile nodes to estimate their locations. The Monte-Carlo Localization [1] provided simulation based

* The work is supported by a grant from the National High Technology Research and Development Program of China (863 Program) (2006AA01Z222).

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 524–535, 2007.

solutions to anticipate the possible location distribution next step of a sensor based on the movement pattern model and the history of the node's movement. The range-based Sequential Monte Carlo Localization Method [3] uses both two types of range-free and range-based information as well as mobility to obtain accurate position estimations. These simulation based localization techniques take advantage of mobility to improve accuracy and reduce the number of anchors required. But they both suffer one problem: high computation cost.

The energy consumption of localization and the memory size restrictions of sensor nodes limit the use of these simulation based methods. In spite of the improved localization precision, drawing enough samples will be a long time that can easily drain a lot of energy from a sensor node. In this paper, we design an energy-efficient dynamic simulation based localization algorithm. It uses range measurement information to restrict sample region and establishes a dynamic filtering mechanism to improve the localization performance and efficiency. It makes more effective predictions to increase its robustness even when no messages are received.

The rest is organized as follows. After the problem formulation in Section 2, we introduce our localization algorithm in Section 3. The algorithm is analyzed in Section 4. Section 5 reports on simulated experiments and compares our results with other localization techniques. Section 6 summarizes the conclusions.

2 Problem Formulation

We assume that the time is divided into discrete time units. In a time unit, a node should update its current location with all the information available even if it keeps static temporarily.

The mobile localization problem can be stated in a state space form as follows. Let t be the discrete time, o_t denotes the observations of a node received from anchors. l_t denotes the position distribution of the node at time t. It consists of a collection of N valid samples after the recursive calculation. A transition equation $p(l_t|l_{t-1})$ describes the prediction of node's current position based on previous position, and an observation equation $p(l_t|o_t)$ describes the likelihood of the node being at the location l_t given the observations. Our study focus on the efficiency and accuracy of localization algorithm, which means it should process all the information above availably and calculate more accurate coordinates in less time.

3 Dynamic Simulation Based Localization Algorithms

Motivated by Baggio's work in [2] and Dil's work in [3], we use the node's connectivity constraints and range measurement information to calculate the sample region in prediction phase. It significantly improves the effectiveness of samples and thus reduces computational complexity in filtering phase. In addition, we establish a dynamic filtering mechanism based on the area of sample region and the velocity prediction to anticipate the posterior distribution of possible locations using a set of weighted samples. With the available samples and associated weights, we can finally compute a refined position estimation.

3.1 Pseudo Codes of the Algorithms

We are interested in performing localization in a more general network environment. We assume that all sensors can move in the monitored area at any direction and at any speed or keep static temporarily. But a sensor node typically has little or no control of its mobility, and is unaware of its speed and direction. Moreover we assume that the maximum speed is unknown. In consideration of the existence of a large number of mobile nodes and a few static nodes, localization algorithm should not only be applicable for mobile nodes, but also make a judgment for those static nodes.

```
For every “time unit” do
    Receive, save and forward messages         (section 4.1)
    Compute sample region                      (section 3.2)
    If sample region is small enough
       Then predict in the region              (section 3.2)
       Compute weight for predictions          (section 3.3)
    Else
        Predict and use dynamic filtering      (section 3.3)
       Compute weight for each saved prediction (section 3.3)
    Compute velocity prediction                (section 4.2)
    Compute final position                     (section 3.3)
End For;
```

This shows an overview in pseudocode of the algorithm. The nodes locally use this algorithm to estimate their positions with the received information of the anchors and neighbor nodes. The different phases of the algorithm are discussed in the following subsections.

3.2 Sample Region

In each time unit, every node would hold a list which consists of the location of anchors n-hop away and associated range measurement to them. Given range measurement rm to anchor position a, we can make a judgement about the node's current location. It is likely to locate either at the edge of the circle with origin a and radius rm or within the region of the circle because of range measurement errors. Thus, we can easily build the sample region by overlapping all the associated circles as follows.

The rectangle in Fig.1 is the sample region. We build it simply by calculating coordinates (x_{min},x_{max}) and (y_{min},y_{max}) as follows, with (x_j,y_j) being the coordinates of the considered anchor j, n being the total number of anchors heard and rm_j being the associated range measurement information.

$$
\begin{aligned}
x_{\min} &= \max_{j=1}^{n}\left(x_j - rm_j\right) \quad x_{\max} = \min_{j=1}^{n}\left(x_j + rm_j\right) \\
y_{\min} &= \max_{j=1}^{n}\left(y_j - rm_j\right) \quad x_{\max} = \min_{j=1}^{n}\left(y_j + rm_j\right)
\end{aligned}
\tag{1}
$$

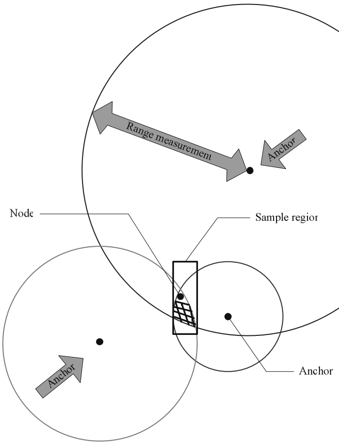


Fig. 1. Build Sample Region

Hence, the samples are drawn randomly within the region. However, Fig. 1 shows discrepancy between the actual area overlapped (shaded area) and our calculated region. So we keep a filtering step (Section 3.3). But it's not necessary when the calculated region is small enough. In this case, for the sake of energy saving and efficiency, we assume that all the predictions are effective.

In prediction phase, regardless of the set of possible locations computed in the previous step (l_{t-1}) and the mobility model, we make predictions merely on the basis of information received. Good samples can be drawn as long as the node is well-connected, even if the node is static temporarily.

3.3 Dynamic Filtering and Weights

In this step, the node filters the impossible locations based on new observations and mobility. It will directly affect the performance of localization algorithm. We establish a dynamic filtering mechanism based on the area of sample region and the velocity prediction to reduce computational time.

Because the transmission range is modeled as a perfect circle and only four-hop away range measurement information is available, the basic filter condition holds:

$$
filter(l) = \forall s \in S, d(l,s) \le rm_S \ \wedge \ \forall s \in T, r < d(l,s) \le rm_T \tag{2}
$$

S is the set of one-hop away anchors, T is the set of four-hop away anchors; rm_S , rm_T are associated range measurement and $d(l,s)$ is the distance between prediction l and anchor *S*. Though range measurements between nodes contain some error, it's also a more effective modified filter condition than using the transmission range.

The transition equation $p(l_t|l_{t-1})$ is often used in prediction phase of those known simulation based localization schemes to anticipate the locations.

$$p(l_t|l_{t-1}) = \begin{cases} \frac{1}{\pi v_{\max}{}^2} & if\ d(l_t|l_{t-1}) < v_{\max} \\ 0 & if\ d(l_t|l_{t-1}) \geq v_{\max} \end{cases} \tag{3}$$

But it's obviously incorrect when the velocity is much less than the maximum speed. Thus, it will lead to much more time and energy in filtering the impossible predictions for their deviation against connectivity constraints.

We use a modified transition equation as our added filter condition with L_{t-1} being the set of possible locations computed in the previous step and *v* being the current velocity between time *t-1* and time *t*.

$$filter(l) = \forall a \in L_{t-1}, d(l,a) \leq v \cdot t \tag{4}$$

In our algorithm we use a velocity prediction instead of actual velocity since there is no velocity measurement hardware. The velocity prediction ,$v_{prediction}$, may contain some errors since we compute it based on L_{t-1} and L_t.

Thus, we adopt a dynamic filtering mechanism that uses both the two filter conditions above (Equation 2 and Equation 4) and increases *v* (Equation 4) gradually in iterative computation. It uses different filter functions to increase its robustness even when high velocity prediction errors are present.

Besides, after this filtering phase, a weight is estimated for each prediction in L_t. With the available predictions and associated weights, we can finally compute the node's position using a weighted mean method as in [3]. The calculation of weights will conform to the following principles :

- It is based on the computation load when the dynamic filtering is used. The more times of iterative computation it takes to draw a good sample, the smaller its associated weight is.
- Without using the dynamic filtering, all samples in Lt use the same weight.

4 Analysis

In this section we analyze the message propagation, velocity prediction and observations used in our algorithm. We conclude by discussing the possibility of dynamic localization control with our algorithm.

4.1 Messages

In mobile sensor networks, mobility gives some obvious significances and potentials. It makes the sensor networks better acquire information [6]. Meanwhile, more available messages propagated can lead to better performance of localization. The way we propagate the information and process the messages received will directly affects the computational cost and the performance of localization. In our algorithm, we focus on two kinds of messages: the coordinates information spread and the associated range measurement information:

1. The coordinates information spread. An anchor spreads its current location in each announcement and neighbor nodes can transmit information about anchor locations (the set of all anchors and their locations heard). To make full use of collaboration in sensor networks where the anchor density is low and the node distribution is irregular, the Time-To-Live (TTL) of the messages which indicates the number of times a message is forwarded is kept by 4 in our algorithm.
2. The range measurement information. Our algorithm uses sum-dist, nameless in [4] and later named in [9], to help the nodes determine their distance to one or multiple anchors. We use this most simple distance-determining solution to restrict the sample region.

   Thus, we can design a simple format for information propagation as follows.

$$\begin{array}{ll} Beacon : S \rightarrow \mathrm{Re}\,gion & \mathrm{Hello} \mid \mathrm{ID_S} \mid loc_t \mid TTL \\ Node : N \rightarrow \mathrm{Re}\,gion & \mathrm{Hello} \mid \mathrm{ID_N} \mid \left\{\left(\mathrm{ID_S}, loc_{S_t}, rm\right)\right\} \mid TTL \end{array} \tag{5}$$

For most of simulation based localization techniques, a wealth of information can significantly improve localization performance. It makes the sample region and the filter conditions available more restricted. Therefor, it is likely to reject samples more often in the filtering phase, increasing thereby the number of iterations the algorithm needs to fill the sample set entirely. To reduce the computation cost and avoid meaningless energy consuming, our algorithm sets a bound on the area of sample region. When the area of sample region is less than the bound, we make predictions only with the basic filter condition (Equation 2). Once the area of sample region is small enough (as pointed out in Section 3.2), we assume that all the samples are effective enough. In this case, the filter phase is unnecessary.

4.2 Velocity Prediction

Known simulation based localization techniques often assume that nodes have a known maximum velocity v_{max} which is uniform for all nodes at any time, while we considering a more general network environment. But in a mobile sensor network, the velocity is obviously different among nodes. Since there are a few static nodes, using a fixed maximum velocity may be insufficient if the sensor is moving faster than this maximum velocity. Conversely, if the sensor is not moving fast or keep static temporarily, using the maximum velocity for prediction may be overly aggressive.

In our algorithm, we anticipate the node's current velocity based on the L_{t-1} and L_t , being the velocity estimation for next time step. Though it may contain some errors, it is only used by a single node. Different nodes may have different velocity predictions, thereby making the algorithm suited for heterogeneous network environment.

In addition, the velocity prediction is update periodically. Thus, we can make a judgement about a node's mobility. The algorithm changes the filter conditions (described in Section 3.3) as long as the node changes its velocity.

4.3 Observations

It is possible that a node receives nothing while there are no available neighboring nodes. Thus, no valid filter conditions can be made because of lack of information. In this case, we make predictions without filtering as follows.

$$\mathrm{p}(1_{\mathrm{t}} | 1_{\mathrm{t-1}}) = \begin{cases} \dfrac{1}{\pi v_{prediction}{}^{2}} & \textit{if } \mathrm{d}(1_{\mathrm{t}} | 1_{\mathrm{t-1}}) < v_{prediction} \\ 0 & \textit{if } \mathrm{d}(1_{\mathrm{t}} | 1_{\mathrm{t-1}}) \geq v_{prediction} \end{cases} \tag{6}$$

The longer time a node has nothing received, the less accurate the estimation will be. In addition, once the node has observations again, it is possible that all predictions based on the last nothing-received time step will be rejected by the new filter conditions. So the algorithm makes predictions as if it had no previous predictions. This increases the performance of the algorithm in several ways:

- The samples in set L_{t-1} are made under the situation of lack of information. They are likely not to satisfy the new connectivity constraints and will lead to infinite computation. Making estimation merely based on new observations can avoid infinite computation, thereby saving energy for nodes.
- The positioning error decreases rapidly as time goes by if there are no messages available. Making estimation based on new observations gives a better representation of the position distribution.

5 Simulation Results

In this section, we use the ns-2 discrete event simulator to evaluate the DSL algorithm by measuring how its estimated location errors and computational cost vary with various network and algorithm parameters described in Section 5.1.

In addition, we compare our results to other localization techniques: these are the Range-free SMCL scheme [1] , MCB [2] and the Range-based SMCL scheme [3].

5.1 General Simulation Set-Up

For all of our experiments, sensor nodes are randomly distributed in a 300m×300m rectangular region. We assume a fixed transmission range , *r* ,of 50m for both nodes and anchors. The frequency for localization is set at a fixed rate.

We adopt the random waypoint mobility model [8] for both nodes and anchors in most of Scenarios. In Section 5.5, we use the BonnMotion tool to generate other two kinds of mobility scenarios: Gauss-Markov [8] and RPGM [8] . The parameters we set or vary are:

- The number of predictions drawn by the sampling function. In general, we use a number of 50 samples.
- The number of nodes and anchors placed within the area. In general, we use a number of 115 nodes and 11 anchors. The general set-up has a node density (average number of 1-hop away nodes) of: 10 and an anchor density of: 1.
- The speed of the nodes, which we choose randomly from [$0, v_{max}$]. The node speed is given as a ratio of the transmission range. In general, we use a speed range of [0, r].
- The Time-To-Live (TTL) of the messages. This value indicates the number of times a message is forwarded. In general, we use a TTL of 4 for every scenario.
- Pause time. In the random waypoint model, a node randomly chooses its destination, speed and pause time after arriving at the destination. In general, the pause time is set to 0. But in Section 5.3, we test our algorithm with different values of pause time.
- We tested each simulation setup for 10 runs, each consisting of 50 time units.

5.2 Accuracy and Cost

The key metric for evaluating a localization technique is the accuracy of the location estimates versus the communication and computational cost. In this section, we compare our results to other simulation based localization techniques: the Range-free SMCL scheme, MCB and the Range-based SMCL scheme.

In Fig. 2 and Fig. 3 we show the accuracy comparison between the four localization techniques under two kinds of environmental settings: $v_{max} = r$ and $v_{max} = 0.2r$. The DSL algorithm we proposed significantly outperforms the two known range-free simulation based algorithms: MCL and MCB, with error decreased by an average of 50%. The two range-free localization solutions performs poor because they depend on higher anchor density when a TTL value of 2 is used.

Intuitively, higher density of node and anchor will lead to better localization performance. We compare different computation cost for the range-based SMCL scheme and our algorithm because the same value of TTL is used. Fig. 2 and Fig. 3 illustrate that both of algorithms using range measurement information has a similar

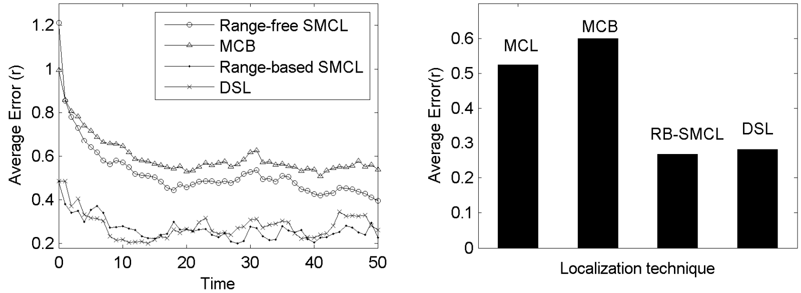


Fig. 2. Accuracy Comparison. nd =10, Sd =1, $v_{max} = 0.2r$.

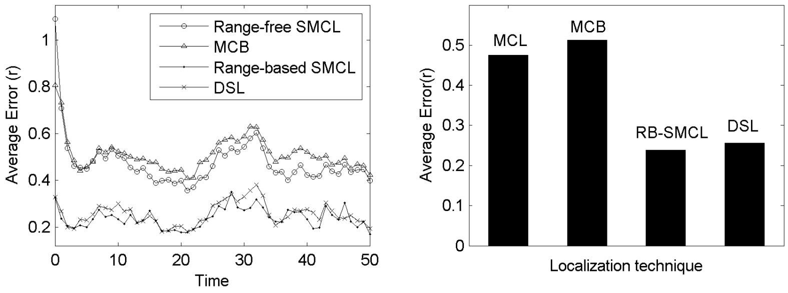


Fig. 3. Accuracy Comparison. *nd* =10, *Sd* =1, $v_{max} = r$.

high localization accuracy. But their processing time of simulation in ns-2 is quite different in the same environmental settings. Our simulation results show that the processing time reduces by an average of 91% for a similar high localization accuracy.

The range-based SMCL algorithm takes much more time mainly because it takes every range measurement as a sampling function. This could easily lead to a tedious process to estimate accurate location while a node receives plenty of information, draining thereby a lot of energy from nodes. In our algorithm, we process all the information availably, for example set a bound on the area of sample region, to reduce the computation cost and avoid meaningless energy consuming. The significantly reduced processing time indicate that DSL is an energy-efficient localization algorithm which improve the computation cost and the energy consumption.

5.3 Pause Scenario

In this section, we test our algorithm with different values of pause time. As pointed out in [11], the random waypoint model suffers from the decay of average speed. It may get worse when considering the pause time, shown as follows.

The maximum speed is set to *r*. In Fig. 4, the average speed of all nodes (calculated by ns-2) decreases rapidly, varying from 0.2 *r* to 0.06 *r*, when the maximum pause time varies from 5s to 40s. The average speed keeps in a very low degree because a node

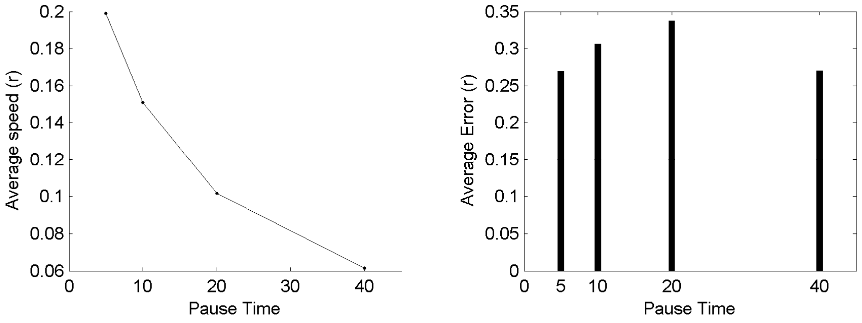


Fig. 4. Pause Scenario

keep static temporarily and the velocity is quite different among nodes. But the average estimate error keeps in a low degree, varying from 0.27 *r* to 0.34 *r*, which shows the DSL algorithm is adaptive for large speed-heterogeneous sensor networks.

5.4 Motion Models

In this section, we test our algorithm with other mobility models: the Gauss-Markov model [8] and the Reference Point Group Mobility model (RPGM) [8] .We use the BonnMotion tool to generate different Scenarios under this two motion models.

One of problems for mobile localization schemes in literature is that they impractically simplify the mobility pattern and thus they do not work well when nodes move randomly [10]. Though the Gauss-Markov Mobility Model deviates from the linear pattern, our algorithm performs well with an average estimate error of 0.34*r*, in both fast and slow speed, shown as follows.

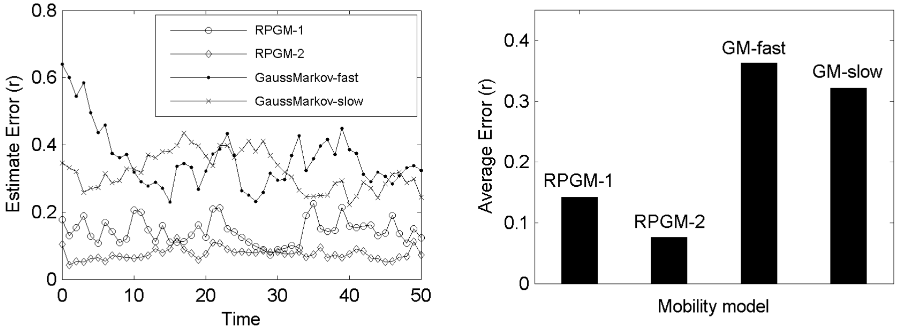


Fig. 5. Different Mobility Models. n_d =10, S_d =1, v_{max} = *r* or 0.2*r*.

For RPGM, we put all nodes and anchors into single or two groups and set the maximum speed v_{max} = *r*. We assume there is a boundary in the network so nodes cannot move out. Fig. 5 shows the location accuracy when we keep the maximum group motion speed at r per time unit and vary the group sizes. The estimate error decreases when the group size increases, varying from 0.15 *r* to 0.08 *r*. Since all nodes are moving in the same way, the more groups are divided, the more numbers of useful new observations are received. Thus, it performs better.

The simulation results show that our algorithm is adaptive. It is primary based on the amount of information received and is independent of mobility models. It performs well in both linear and non-linear mobility pattern.

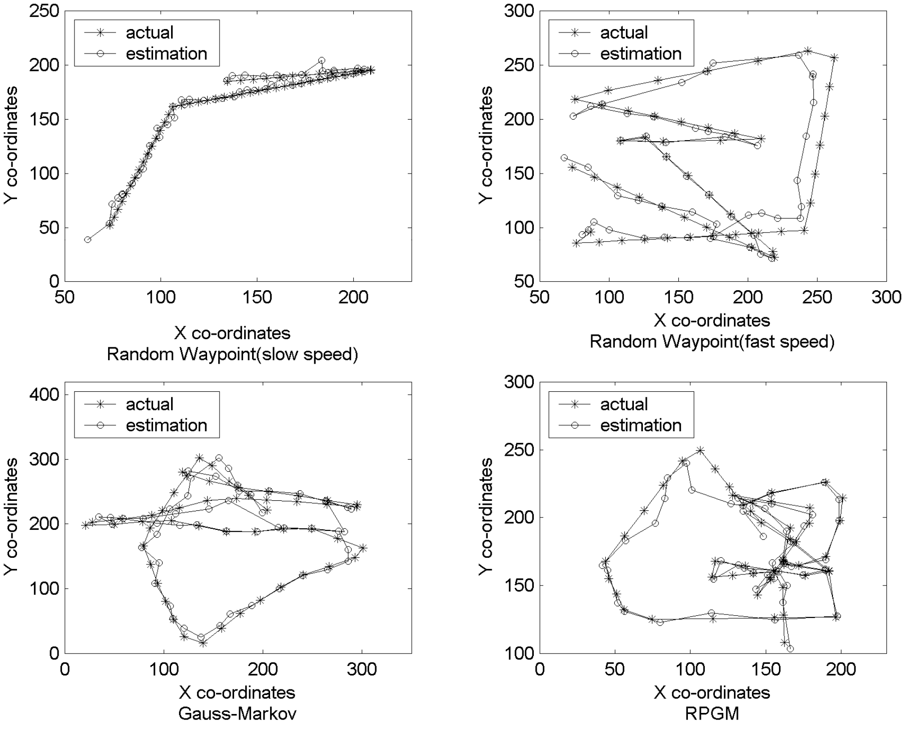
5.5 Trajectory

In this section, we test our algorithm by comparing the node's actual movement trajectory with the estimation trace in different kinds of mobility models and network environment.

In Fig. 6 we show the trace results for four different well-connected nodes in different scenarios: the random waypoint with slow speed, the random waypoint with high speed, the Gauss-Markov with high speed and the RPGM with high speed.

As shown in the figure, the deviation between two trajectories is small and the average estimate error is no more than 0.1 *r*.

Navigation of mobile wireless sensor networks and fast target acquisition without a map are two challenging problems in search and rescue applications [12]. In Fig. 6 we also show that, with enough messages available, the DSL algorithm may has a good performance in mobile tracking under different settings of environment.



Random Waypoint(slow speed)

Random Waypoint(fast speed)

Gauss-Markov

RPGM

Fig. 6. Node trajectory comparison in different Scenario

6 Conclusions

In this paper, we proposed a novel dynamic simulation based localization algorithm for mobile sensor networks. We use all information which consists of range measurement and connectivity constraints to improve localization accuracy and energy efficiency. The simulation results show that our algorithm outperforms the best known simulation based localization schemes under a wide range of conditions:

– Localization accuracy is improved by an average of 24%.

- The processing time reduces significantly for a similar high localization accuracy in the same environmental settings, reducing thereby localization energy to a great extent.
- The algorithm is adaptive for different kinds of mobility models and large speed-heterogeneous sensor networks.

References

1. Hu, L., Evans, D.: Localization for mobile sensor networks. In: MobiCom 2004. Tenth International Conference on Mobile Computing and Networking, Philadelphia, Pennsylvania, USA, pp. 45–57 (September 2004)
2. Baggio, A., Langendoen, K.: Monte-Carlo Localization for Mobile Wireless Sensor Networks. In: 2nd International Conference on Mobile Ad-hoc and Sensor Networks 2006 Mobile Ad-Hoc and Sensor Networks, Proceedings, pp. 317–328 (December 13-15, 2006)
3. Dil, B., Dulman, S., Havinga, P.: Range-Based Localization in Mobile Sensor Networks. In: Römer, K., Karl, H., Mattern, F. (eds.) EWSN 2006. LNCS, vol. 3868, pp. 164–179. Springer, Heidelberg (2006)
4. Savvides, A., Park, H., Srivastava, M.: The Bits and Flops of the N-Hop Multilateration Primitive for Node Localization Problems. In: First ACM International Workshop on Wireless Sensor Networks and Application, Atlanta, GA (September 2002)
5. Tilak, S., Kolar, V., Abu-Ghazaleh, N.B., Kang, K.: Dynamic localization control for Mobile Sensor Networks. In: IPCCC 2005. Conference Proceedings of the 24th IEEE International Performance, Computing, and Communications Conference, pp. 587–592 (2005)
6. Yuan, L., Chen, W., Xi, Y.: A Review of Control and Localization for Mobile Sensor Networks. In: WCICA. Proceedings of the World Congress on Intelligent Control and Automation, pp. 9164–9168 (2006)
7. Doucet, A., Godsill, S., Andrieu, C.: On Sequential Monte Carlo Sampling Methods for Bayesian Filtering. Statistics and Computing 10, 197–208 (2000)
8. Camp, T., Boleng, J., Davies, V.: A Survey of Mobility Models for Ad Hoc Network Research. Wireless Communications and Mobile Computing 2(5), 483–502 (2002)
9. Langendoen, K., Reijers, N.: Distributed localization in wireless sensor networks: A quantitative comparison. In: Computer Networks (Elsevier), special issue on Wireless Sensor Networks (2003)
10. Al-laho, M.Y., Song, M., Wang, J.: Mobility-Pattern Based Localization Update Algorithms for Mobile Wireless Sensor Networks. In: Jia, X., Wu, J., He, Y. (eds.) MSN 2005. LNCS, vol. 3794, pp. 143–152. Springer, Heidelberg (2005)
11. Yoon, J., Liu, M., Noble, B.: Sound Mobility Models. In: MOBICOM. Models. Proceedings of the Annual International Conference on Mobile Computing and Networking, pp. 205–216 (2003)
12. Zhang, Q.Q., Sobelman, G, He, T.: Gradient-driven target acquisition in mobile wireless sensor networks. In: Mobile Ad-Hoc and Sensor Networks, Proceedings: 2nd International Conference on Mobile Ad-hoc and Sensor Networks, pp. 365–376 (December 13-15, 2006)

SLTP: Scalable Lightweight Time Synchronization Protocol for Wireless Sensor Network

Sepideh Nazemi Gelyan[1], Arash Nasiri Eghbali[2], Laleh Roustapoor[2],
Seyed Amir Yahyavi Firouz Abadi[3], and Mehdi Dehghan[2]

[1] Dept. Of Computer Engineering, Islamic Azad University, Firouzkooh Branch
[2] Computer Engineering Department, Amirkabir University of Technology
[3] Electrical & Computer Engineering Department, University of Tehran
s.nazemi@iaufb.ac.ir,
{eghbali, roustapoor, dehghan}@aut.ac.ir,
a.yahyavi@ut.ac.ir

Abstract. In wireless sensor networks, time synchronization is a critical problem. In this paper, we propose SLTP, a Scalable Lightweight Time-synchronization Protocol for wireless sensor networks. By using passive clustering and linear regression SLTP can reduce the energy consumption of network nodes and also decrease the overhead of creating and maintaining the clusters. Moreover SLTP uses linear regression to compute the time. Therefore, it can calculate the clock skew and offset between each node and its cluster head in order to estimate the local time of remote nodes in the future or the past. Simulation results show that by this we can gain considerable improvements in power consumption, accuracy and scalability in comparison to similar algorithms.

1 Introduction

Recently a new kind of wireless networks has emerged, one that has cheaper and smaller nodes [1]. These nodes are sensors which gather and process information from the physical and real world. They are used for geophysical monitoring and observing the special environments such as war zones, wild life, etc. Special and strategic applications of these networks in inaccessible and dangerous environments can cause some restrictions for network managers. For instance, the managers are unable to access nodes for recharging, reconfiguring and reprogramming. So when the nodes are damaged or their batteries are discharged, they are neither recyclable nor reprogrammable [8].

In sensor networks, the main reason for nodes failure is the discharge of batteries. Energy efficiency is a critical issue in wireless sensor networks [12]; therefore, using energy efficient programs and algorithms on these nodes is of great significance [12]. One of the most important problems in computer networks, including wireless sensor networks, is time synchronization among nodes. In view of the fact that wireless sensor networks are used for monitoring strategic environments, the accuracy of the information gathered is crucial. Information accuracy highly depends on time [8]; as a result, time synchronization plays an important role. Traditional time synchronization

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 536–547, 2007.

protocols are not suitable for sensor networks [12, 8]. For example NTP consumes too much energy in order to exchange the required messages and requires using GPS which is costly for sensor nodes [8].

Over the past few years, many time synchronization algorithms have been suggested. The main goals of these algorithms were to increase the time synchronization accuracy and to decrease the power consumption. RBS [1] is one of the most cited protocols in this field. It synchronizes a set of receivers by using linear regression; in other words, RBS presents a way that enables nodes to calculate local time of remote nodes in proportion to their local times. This protocol is not scalable and in case of increase in the number of nodes its efficiency decreases significantly. Although RBS presents a solution for increasing the accuracy, it extremely suffers the network collisions.

Some algorithms such as TPSN and LTS [3, 2] use tree construction. In general, in these algorithms, increase in the depth of the tree results in higher error rates. Also the creation and maintenance of the tree causes high overhead. In addition, tree-scanning algorithms require having some global information about the network which is inconsistent with distribution concept; indeed, wireless sensor networks are distributed systems that do not have access to such information [12, 2]. Some algorithms like PCHTS [6] and PCTS [5] use clustering techniques. PCHTS suffers high overhead due to costly creation and maintenance of the clusters.

PCTS uses the passive clustering concept, which results in efficient energy consumption; on the other hand, it frequently has to set the node's clock to calculated values which leads to high-energy consumption by CPU. These values are calculated using the averaging technique. By using passive clustering, our proposed protocol not only decreases the nodes energy consumption but also reduces the overhead of creation and maintenance of the clusters. Moreover, we use linear regression for computing skew and offset for clock of each node in proportion to its cluster head. This helps nodes to estimate the local time of remote nodes in the future or the past.

We used NS2 for simulating and evaluating our model. The organization of this paper is as follows: We first review related works in Section 2. In Section 3, first we discuss passive clustering (3.1) as a basis for our method; afterwards we will describe our protocol in detail. In section 3.4, we demonstrate the theoretical analysis of our method. Error analysis is discussed in section 3.5. Section 4 presents our simulations' setup and results. Protocol overhead is discussed in section 4.3.2. Finally, we offer our conclusions and describe our plans for future work in section 5.

2 Related Works

RBS [1] synchronizes a set of receivers by using the available broadcasting channel in the network physical layer. The reference node broadcasts a predefined number of synchronization packets to its neighbors and these packets are received approximately at the same time. At this time receivers record their local times. After that, they exchange the received times with their neighbors. Next, they use linear regression for computing clock offset and skew corresponding to their neighbors.

RBS does not readjust the clock. Also it does not consider the nondeterministic errors such as send time delay and access time delay, since broadcasting creates the same amount of these errors for all nodes. The most important disadvantage of RBS is that it is not scalable. In other words, by increasing the number of nodes, synchronization accuracy falls and the number of exchanged messages for synchronization increases.

LTS [2] creates a low-depth spanning tree composed of the nodes in the network, and uses a pair-wise synchronization algorithm for each one of the two nodes on the same edge of the tree [2, 3]. Reference node starts time synchronization and it continues until all leaves are synchronized. LTS uses Breadth-first-search algorithm which has higher communication overhead compared to other tree-construction algorithms. Also executing BFS in distributed systems is difficult.

TPSN [3] has two phases. At the first phase, level discovery phase, a tree is created. At the second phase, the synchronization phase, each node in i^{th} level synchronies itself with the node in $(i-1)^{th}$ level by using a pair-wise synchronization algorithm.

TSYNC [4] uses Multi channel nodes. This attribute results in lower collision rates. Each node has two channels, control and clock channel. All nodes use the same control channel but clock channel is unique for each node. Algorithm has two protocols. One of them is HRTS which is used for synchronizing the whole network and the other one is ITR which lets each node to synchronize itself on-demand. Each node that wants to be synchronized sends an ITR request message to its parent and this is repeated until the request message reaches the base station. The base station returns its clock though the clock channel to the requesting node. TSYNC uses broadcasting like RBS, but it has less overhead than RBS. Disadvantage of this algorithm is that we made an assumption about having multi channel sensor nodes which is not always the case.

CHTS [6] uses clustering technique and nodes are able to change their radio ranges. Network is not homogenous, some nodes are high performance, and some are low performance. Cluster heads are selected from high performance nodes. CHTS has three sub algorithms; cluster head tree construction, cluster member tree construction, and the synchronization algorithm. CHTS makes too many assumptions in order to select cluster heads and cluster members. Moreover, algorithm extremely suffers from collisions, if the number of nodes increases.

PCTS [5] uses passive clustering and overhearing in time synchronization phase. It has two phases. At first cluster heads collect clock information from members of their clusters. Next, they calculate the average value and send the calculated clock to members of their clusters. Some nodes that are members of more than one cluster have to calculate the average clock of all of their cluster heads and send the calculated clock to them. This algorithm has to set clock continuously which results in more power consumption and in order to do this it has to send lots of time synchronization packets which in turn causes a lot of collisions, during the time synchronization phase.

3 Protocol Description

3.1 Review on Passive Clustering

Passive clustering is a method for creating clusters with less overhead than other methods [14]. Its setup time is very short and clustering the network is fast. The most eager node to become “cluster head” starts this procedure [15]. In our implementation this node broadcasts a special packet with a flag. Flag’s value can be either ø or 1 and the initial value is ø. The nodes that receive this packet will become "cluster members" and change the flag to 1, and then rebroadcast it. Adjacent nodes that receive it in turn will become "cluster head" and change the flag to ø and then broadcast too. This procedure is repeated until the whole network is scanned. During this procedure, some nodes receive packets from two or more cluster heads. These nodes change their status to "gateway" [5, 15]. We will discuss passive clustering in more detail in section 3.3.1.

3.2 Assumptions

Time synchronization is done between cluster members and their cluster heads based on the assumption that cluster members do not need to exchange messages with each other, because analyzing data and query processing is done by cluster heads. However, time synchronization among cluster members of the same cluster or different clusters is possible. The nodes have the same abilities such as power level, radio range, etc. Nodes are able to change their status to "cluster head", "cluster member" or "gateway".

Our algorithm does not require nodes to readjust their clocks. The use of linear regression for calculating drift between nodes' clocks enables them to estimate the time of one another. This method has two advantages [12]:

1. Clock oscillators all tick at different rates resulting in different clock disciplines. We let each clock to work with its discipline.
2. Clocks do not require continuous readjustments by CPU which is important for efficient energy consumption.

We do not employ a global time in our algorithm since having only local times is sufficient and more suitable for sensor networks applications [1, 13]. However, using a global time like UTC is also possible.

3.3 Scalable Lightweight Time Synchronization Protocol

Our Algorithm has two phases: configuration phase and synchronization phase. We will describe them below.

3.3.1 Configuration Phase

The intent of this phase is to determine the cluster heads. The procedure for selecting cluster heads influences on network's lifetime [10]. There are many ways to select cluster heads [11]. We use a method close to passive clustering with a little modification so that it fits our needs. Configuration phase is divided into two parts based on network being static or dynamic.

Static mode: The most eager node to become cluster head changes its status to "cluster head" and broadcasts a packet with a Boolean flag. Flag's value is set to ø. Every node that receives this packet changes its status to "cluster member" and sets the packet's flag to 1 and rebroadcasts it. Nodes that have received this packet change their status to "cluster head". They set the packet's flag to ø and broadcast it again. This is repeated until the whole network is covered. Each node stores its cluster head address. Nodes that receive this packet from 2 or more cluster heads will become “gateway”. Gateways must send an ACK message to introduce themselves to their cluster heads. This ACK message contains the list of their cluster heads. Sending an ACK message enables cluster head to select one gateway between clusters. Having one gateway simplifies the routing. We used the following four rules in our implementation of passive clustering:

1. If a cluster member receives the packet from another cluster member, it must not rebroadcast the packet.
2. If a cluster head receives the packet from another cluster head, it must change its status to “cluster member” and rebroadcast the message with flag value 1.
3. If a cluster member receives the packet from another cluster’s head or vice versa, it will not rebroadcast the packet.
4. If a node does not receive any packets for a specified amount of time, it must set its status to “cluster head” and broadcasts a message with flag value ø.

Dynamic mode: In dynamic mode, only the selection of the cluster heads is done. Cluster members and gateways are determined in synchronization phase. Because of the network dynamics, a node that was a gateway in the past may not be one now and a node that belonged to cluster i may now belong to cluster j.

We believe, our algorithm performs relatively well in dynamic networks. In case of applications that the rate of change in network structure is faster than configuration phase’s speed, our algorithm may not work well. In dynamic networks configuration procedure must be executed before doing time synchronization. This procedure helps in uniform distribution of cluster heads.

3.3.2 Synchronization Phase

After configuration phase is finished and cluster heads are selected, synchronization phase starts. During a defined time period which we call “setup time” cluster heads start broadcasting some packets in random intervals. These packets contain local times of cluster heads. When one of these packets is received by a cluster member, the cluster member records its own local time. In dynamic mode by receiving a packet, cluster members can figure out which cluster they belong to.

By using linear regression method each cluster member can calculate its clock offset and clock skew in proportion to its cluster head. In dynamic mode, if a node receives packets from two or more cluster heads, it will become gateway. In general, gateway’s task is to convert local time of cluster heads to each other.

In static mode, time synchronization is repeatedly done in specific time intervals. If a node needs to synchronize itself with another node during this period, it can broadcast an independent time synchronization request message. The first cluster head that receives this message starts synchronization procedure and broadcasts synchronization packets.

In dynamic mode as mentioned before, we have to execute configuration phase before each synchronization phase, in order to have a uniform distribution of cluster heads. If during synchronization phase a node does not receive any packets, node changes its status to “cluster head” and starts broadcasting synchronization packets with its local time. If the network's distribution is uniform, we may hope that this node will be able to connect through one gateway to the network.

Below is the pseudo-code of our algorithm:

```
CH: Cluster Header; CM: Cluster Member
G: Gateway; $h_i$: local time of node i

Program timesync (output);
Begin
  Configuration (base station); {in this part CHs and CMs are selected by our Passive Clustering algorithm}
  For each $n_i$ in CH set do broadcast (ID, $h_i$);
  For each $n_i$ in CM set do
  Begin
      Receive (ID, $h_i$);
      Store (ID, $h_i$);
      Compute regression ($h_i$, $h_j$);
      Co=clock offset ($h_i$, $h_j$);
      Cs=clock skew ($h_i$, $h_j$)
  End;
  If ($n_k$ is in the CM set) and ($n_k$ has received from more than one CH)
  Then Begin
      $n_k$.Status:= G;   {add $n_k$ to gateways set}
      For each i that has sent packet to same $n_k$ do
      Compute regression ($h_i$, $h_k$) {i is the member of CH that has sent
    a packet to the same $n_k$}
  End;
  If there is $n_m$ that does not receive any packet from CH then
  Broadcast (ID, $h_m$);
End.
```

3.4 Analysis by Example

To describe our analysis clearly we use some examples. After execution of our algorithm, each node except for the cluster heads calculates the following equation:

$$h_{chi}= \alpha h_{cmi}+ \beta. \tag{1}$$

Which α is the clock skew and β is the clock offset between cluster head and cluster member. Assume that CM_1 at its local time h_{cml} sees an object in location (X_1, Y_1). It calculates the local time of its cluster head, in proportion to its own local time of the object's observation. CM_1 sends this time and the location of the object to its cluster head. This procedure is repeated by CM_2. CM_2 sees the object in its local time h_{cm2} in (X_2, Y_2). Therefore cluster head can calculate the speed and direction of the moving object (figure 1-a).

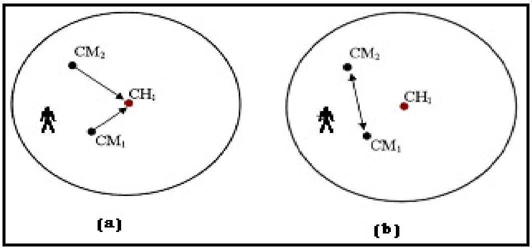


Fig. 1. CH1 receives the observation time of the object from CM1 and CM2, (b) CM1 and CM2 are synchronizing their clocks

If two nodes in the same cluster want to exchange information, calculating their own *α* and *β* is sufficient. By the following equation, they can calculate time of each other (figure 1-b).

$$\begin{aligned} &(1)\ \mathrm{CM}_1 : \mathrm{h}_{\mathrm{CH}_1} = \alpha_1 \mathrm{h}_{\mathrm{CM}_1} + \beta_1 \\ &(2)\ \mathrm{CM}_2 : \mathrm{h}_{\mathrm{CH}_1} = \alpha_2 \mathrm{h}_{\mathrm{CM}_2} + \beta_2 \\ &\therefore \mathrm{h}_{\mathrm{CM}_1} = \frac{\alpha_2}{\alpha_1} \mathrm{h}_{\mathrm{CM}_2} + \left(\frac{\beta_2 - \beta_1}{\alpha_1} \right) \end{aligned} \tag{2}$$

In general we can formulate the relation between local time of two nodes as follows. CM$_2$ can convert its local time to CM$_1$'s local time by using equation 2 (figure 2).

In spite of the fact that this calculations have error, if α and β are calculated accurately in each hop, they would remove each other's effect (Equation 3). As result, we will have fairly constant error amount in multi-hop time synchronization algorithm. In our Implementation it is not necessary for CM1 to have all of parameters (α and β) along the path from CM1 to CM2. The calculation is done in each node in the path until the packet reaches its destination.

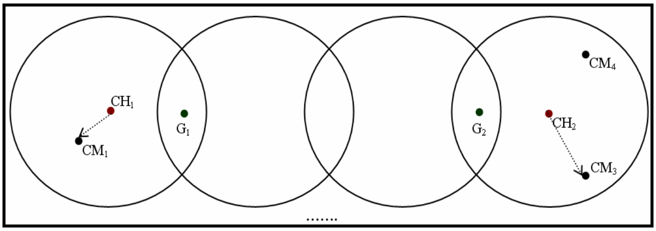


Fig. 2. Two nodes from different clusters are synchronizing their clocks

$$h_{CM_1} = \frac{\prod_{i=1}^{n} \alpha_{2i}}{\prod_{i=1}^{n} \alpha_{2i-1}} h_{CM_2} + \sum_{i=1}^{n/2} \left(\frac{\beta_{2i} - \beta_{2i-1}}{\alpha_{2i-1}} \right) \frac{\prod_{j=1}^{i-1} \alpha_{2j}}{\prod_{j=1}^{i-1} \alpha_{2j-1}} \tag{3}$$

$$\prod_{j=1}^{0} = 1$$

n: Number of nodes in the path between source & destination

3.5 Error Analysis

By broadcasting we can ignore some sources of error in time synchronization [1] such as send time and access time delay. These errors depend on parameters that we cannot control; for instance, the load on the sender's CPU or network's load. By broadcasting, these delays become similar for the whole network and as a result we can ignore them [8, 1]. In addition, the speed of electro-magnetic signal in air is almost *C* (Light speed) [1] and propagation delay is calculated by this equation: *T= x/v, v =c.*

As a result, since sensors have limited range, propagation delay is very small and can be neglected. The remaining error is received time delay, according to [6] we can receive and process the packet in an interrupt. Calculating α and β in each hop has computational error as was mentioned in section 3.4. Therefore, as the path between source and destination node increases the error grows higher.

4 Performance Evaluation

For performance evaluation we used NS 2.31. We selected PC_Avg [5] and RBS [1] for performance comparison and created the same test conditions for all of them during the simulation time. We compare SLTP to PC_Avg in terms of accuracy and scalability. Also, we compare it to RBS for overhead and scalability.

4.1 Simulation Setup

Simulation and evaluation of different algorithms that require different clock configurations on one system are very difficult. To create different clock behavior for each node, we use a different clock drift and offset. The values are selected randomly and each node's clock works according to these parameters. As mentioned in [1], typically two nodes' clock will drift 1-100μsecond per second. Also we assume that maximum clock offset is 1 second. This pessimistic offset should cover for the propagation delay. All nodes have a range of 100 meters and are located in 1000*1000 square meters area. Network topology is grid. Default number of nodes is 140 and default simulation time is 10450s. We run the synchronization algorithm every 1000 seconds. Each cluster head sends 10 initial synchronization packets (which contain its local time) with a 1 second delay between each transmission.

4.2 Time Routing

When a node wants to be synchronized with another one, it has to broadcast a request message. If this node is a cluster member or a gateway, it must send this request to its cluster head(s). The cluster head broadcasts this message to its cluster and saves the address of requesting node as the last hop of this message. Each gateway in this cluster, after receiving this packet, sends it to its entire cluster heads except the one that sent this packet in the first place. Cluster heads that receive this packet will save the gateway's address as the last hop of this message. This procedure is repeated until the packet reaches the destination node. A path is made between requesting node and destination node. In the reverse path in each hop the clock of destination node is converted until it reaches requesting node.

4.3 Simulation Results

Since in each run, different paths between source and destination are selected we decided to run the algorithm 10 times for each different situation and calculate the average of error for each run. If gateways with small clock drifts and skew are chosen along the path, average of error will be smaller.

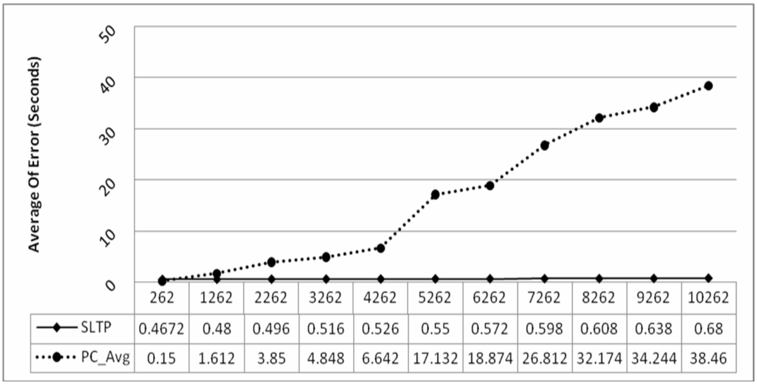
4.3.1 Time Synchronization Results

Figure 3 shows the average of error versus simulation time. It is clear that as time passes, error of SLTP increases gradually, while in case of PC_Avg this increase is dramatic. Since clock of each node works with a different drift and offset, and the difference among clusters' clock can be significant, PC_Avg algorithm will not be able to appropriately synchronize different clusters. Because SLTP uses linear regression, it is able to manage this problem.

Figure 4, shows the influence of number of hops increase on the error. As indicated, SLTP's error in all situations was significantly less than PC_Avg's. Besides, as the number of hops increases, error ratio of PC_Avg goes up dramatically but in case of SLTP this increase is gradual.

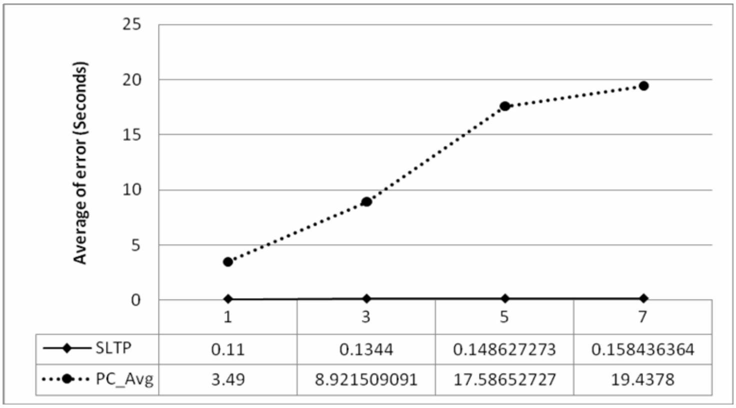
Average of error in SLTP in one hop is 0.13± 0.06s. Increasing the number of hops has a little effect on SLTP's error. The reason for this is mentioned in section 3.4.

Figure 5 compares the scalability of SLTP versus PC_Avg. We can observe that error of SLTP is almost stable and SLTP's error seems independent of the number of nodes. Figure 6 shows the two algorithms' tolerance under different drifts.



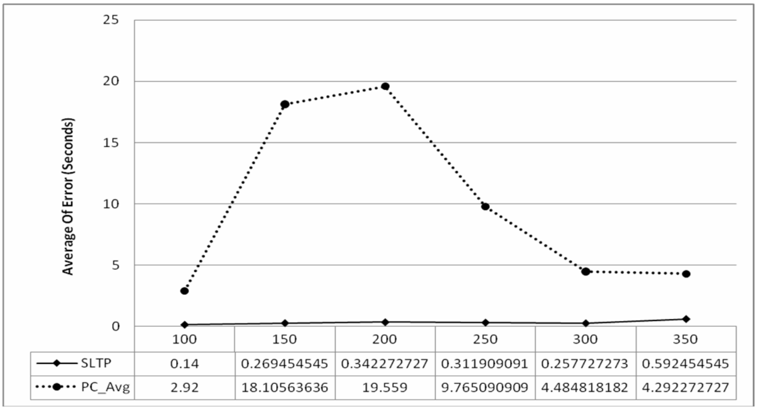
	262	1262	2262	3262	4262	5262	6262	7262	8262	9262	10262
—◆— SLTP	0.4672	0.48	0.496	0.516	0.526	0.55	0.572	0.598	0.608	0.638	0.68
···●··· PC_Avg	0.15	1.612	3.85	4.848	6.642	17.132	18.874	26.812	32.174	34.244	38.46

Fig. 3. Comparison of SLTP's and PC_Avg's error versus simulation time



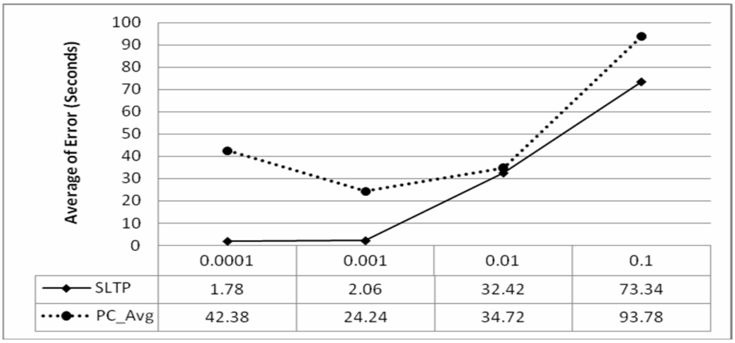
	1	3	5	7
—◆— SLTP	0.11	0.1344	0.148627273	0.158436364
···●··· PC_Avg	3.49	8.921509091	17.58652727	19.4378

Fig. 4. Comparison of SLTP's and PC_Avg's error versus number of hops



	100	150	200	250	300	350
—◆— SLTP	0.14	0.269454545	0.342272727	0.311909091	0.257727273	0.592454545
···●··· PC_Avg	2.92	18.10563636	19.559	9.765090909	4.484818182	4.292272727

Fig. 5. Comparison of SLTP and PC_Avg in scalability



	0.0001	0.001	0.01	0.1
—◆— SLTP	1.78	2.06	32.42	73.34
···●··· PC_Avg	42.38	24.24	34.72	93.78

Fig. 6. Comparison of SLTP's and PC_Avg's tolerance under different drift

4.3.2 Overhead

By using passive clustering and linear regression we decreased the number of exchanged messages in SLTP. Table 1 shows the number of messages used to synchronize the whole network for SLTP and two similar algorithms.

Table 1. Number of transmitted messages for the following algorithms

Number of messages	Algorithm
C * m	SLTP
C*(2 + K)	PC_Avg
N^2 * m	RBS

C: Number of Cluster Heads
K : Number of Cluster Members
N: Number of Nodes
m: Number of Synchronization Packets
C<<K<<N

Note that although the number of cluster members in PC_Avg is usually less than the number of synchronization packets, since all of the cluster members send their clocks to their cluster heads at the same time the possibility of collision and packet loss increases.

5 Conclusion

We introduced the SLTP for time synchronization in wireless sensor network. By using passive clustering and linear regression, not only the accuracy of time synchronization in comparison to similar algorithms is improved but also we were able to reduce the consumption of energy in each node by decreasing the number of exchanged messages. Because of the fact that by expanding the network error doesn't increase, SLTP is scalable. By using linear regression SLTP can tolerate the changes in drift under various situations. SLTP is the best fit for the following situations: When the lifetime of the network is very important, when we need a large network to monitor a wide area, or when our application requires medium time accuracy.

From our point of view SLTP will work in dynamic networks as well as in static networks. In this paper we evaluated SLTP only on static networks. This protocol should be tested on dynamic networks as well.

References

1. Elson, J., Girod, L., Estrin, D.: Fine-Grained Network Time Synchronization Using Reference Broadcasts. In: 5th Symposium on Operating Systems Design and Implementation, vol. 36, pp. 147–163. ACM Press, USA (2002)
2. van Greunen Jan Rabaey, J.: Lightweight Time Synchronization for Sensor Networks. In: WSNA. 2nd ACM International Workshop on Wireless Sensor Networks and Applications, pp. 11–19 (2003)
3. Ganeriwal, S., Kumar, R., Srivastava, M.B.: Timing-sync Protocol for Sensor Networks. In: 1st International Conference on Embedded Networked Sensor Systems, pp. 138–149 (2003)
4. Dai, H., Han, R.: TSync: A Lightweight Bidirectional Time Synchronization Service for Wireless Sensor Networks. In: ACM SIGMOBILE Mobile Computing and Communications Review archive, vol. 8, pp. 125–139. ACM Press, USA (2004)
5. Mamun-Or-Rashid, Md., Hong, C.S., Chi-Hyung: Passive Cluster Based Clock Synchronization in Sensor Network. In: Advanced Industrial Conference on Telecommunications AICT/SAPIR/ELETE 2005, pp. 340–345 (2005)
6. Kim, H., Kim, D., Yoo, S.-e.: Cluster-based Hierarchical Time Synchronization for Multihop Wireless Sensor Networks. In: 20th International Conference on Advanced Information Networking and Applications, pp. 318–322 (2006)
7. Sundararaman, B., Buy, U., Kshemkalyani, A.D.: Clock Synchronization for Wireless Sensor Networks: A Survey. Ad Hoc Networks 3(3), 281–323 (2005)
8. Römer, K.: Time Synchronization and Localization in Sensor Networks PhD thesis. No. 16106, ETH Zurich, Switzerland (June 2005)
9. Blum, P., Meier, L., Thiele, L.: Improved Interval-Based Clock Synchronization in Sensor Networks. In: IPSN 2004. 3th International Symposium on Information Processing in Sensor Networks, pp. 349–358 (2004)

10. Younis, O., Fahmy, S.: Distributed Clustering in Ad-hoc Sensor Network: a Hybrid, Energy-Efficient Approach. In: 23th Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 1, pp. 629–640 (2004)
11. Bush, S.F.: Low-Energy Sensor Network Time Synchronization as an Emergent Property. In: ICCN 2005. 14th International Conference on Computer Communications and Networks, pp. 93–98 (2005)
12. Elson, J., Römer, K.: A New Regime for Time Synchronization. ACM Computer Communication Review (CCR) 33(1), 149–154 (2003)
13. Bandyopadhyay, S., Coyle, E.J.: An Energy Efficient Hierarchical Clustering Algorithm for Wireless Sensor Network. In: IEEE INFOCOM, pp. 1713–1723 (2003)
14. Hać, A.: Wireless Sensor Network Designs. John Wiley & Sons, USA (2003)
15. Handziski, V., Kopke, A., Karl, H., Drytkiewicz, C.F.W.: Improving the Energy Efficiency of Directed Diffusion Using Passive Clustering. In: European workshop on wireless sensor network. EWSN 2004, pp. 172–187 (2004)

Ensuring Area Coverage in Hybrid Wireless Sensor Networks

Nadeem Ahmed[1,2], Salil S. Kanhere[1], and Sanjay Jha[1]

[1] Computer Science and Engineering, University of New South Wales, Australia
{nahmed, salilk, sanjay}@cse.unsw.edu.au
[2] National ICT Australia (NICTA), Australian Technology Park, Sydney, Australia

Abstract. Success of Wireless Sensor Networks largely depends whether the deployed network can provide desired coverage with acceptable network lifetime. This paper proposes a distributed protocol for ensuring area coverage using a combination of mobile and static sensor nodes. Most of the assumptions made in our approach are realistic (sensing model, movement thresholds based on real radio characteristics etc.) and implementable in real-life applications. We demonstrate that, for different type of initial deployments, our proposed movement algorithms consume only 30-40% of the energy consumed by the basic virtual force algorithm. We formulated our problem as Integer Linear Program to arrive at idealistic optimal solutions that form basis of our performance comparison. We validated our results through extensive discrete event simulations.

Keywords: Coverage, deployment, hybrid wireless sensor networks.

1 Introduction

In hostile or harsh environments such as enemy territories in battlefields, fire or chemical spills, it is impossible to deploy the sensor nodes in a predetermined regular topology. Random (possibly aerial) deployment of sensor nodes is a solution in such scenarios. However, such random deployments are highly susceptible to the creation of uncovered regions, referred to as *coverage holes*. There are many factors that contribute to this including the presence of obstacles, sloping grounds like hills, strong winds or dense forestation during aerial deployment, etc. A potential solution for enhancing the existing coverage involves the use of mobility capable sensors [1] [2], which would help fill in the voids. In this paper, we seek to address the problem of determining the current coverage achieved by the non-deterministic deployment of static sensor nodes and subsequently enhancing the coverage using mobile sensors. We propose a distributed *Mobility Assisted Probabilistic Coverage (MAPC)* protocol, which aims at maximizing the area coverage with least expenditure of energy.

Our primary contribution is a pragmatic approach to sensor coverage that we hope would lower the technical barriers to its field deployment. Most of the assumptions made in the proposed protocol are realistic and implementable in

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 548–560, 2007.

real-life applications, e.g., coverage calculations based on realistic sensing model (Section 4), and use of thresholds based on real radio characteristics (Section 5) etc. Further, our movement algorithms result in substantial energy savings (60% to 70% as compared to basic virtual force based algorithms), a major challenge for the success of wireless sensor networks (WSN).

The primary focus of this paper is the design and evaluation of the MAPC protocol. We discuss related research work in Section 2 and provide context to our work in Section 3. We present our protocol and simulation results in Sections 4 and 5. We present our observations and open problems in Section 6.

2 Related Work

Use of mobility capable sensors to guarantee coverage during deployment has been proposed in various research efforts [1], [3], [4], and [5] etc. A potential field based approach is proposed in [1] assuming compact initial concentration of the mobile nodes. Wang et al. proposed three different deployment protocols in [3], that spread out the mobile sensors once coverage holes are detected using Voronoi diagrams. A centralized incremental deployment scheme is proposed in [4] that deploys sensors one by one while requiring line of sight among the nodes. Similarly, a centralized Virtual Force Algorithm (VFA) for the movement of mobile sensor nodes is proposed in [6]. All these works assume an all mobile sensor network. In [5] the coverage problem is solved by a moving robot that is guided by the already deployed nodes for exploring poorly covered areas. Use of mobile sensors for enhancing coverage at deployment stage for hybrid WSN is proposed in [2] and [7]. Wang et al. proposed a bidding protocol [2] where static nodes bid for mobile nodes while mobile nodes also consider the loss of coverage at its existing location due to the movement.

Our work is different from these proposed approaches in several ways. We propose a coverage and energy-aware, distributed variant of the VFA for spreading out the mobile sensors as opposed to the centralized VFA employed in [6]. Our proposal do not require provision of any specialized hardware for determining the range and bearing [1], or line of sight [4]. In addition, we also propose the simulated movement approach for energy efficient movement of mobile sensor nodes. Our approach is different from the proxy based logical movement approach proposed in [7] where static sensor nodes are utilized as proxies for message passing.

3 Protocol Overview

Our system assumes a hybrid network consisting of a large number of static sensors deployed in a non-deterministic manner. We also assume that a few mobile sensors are available for plugging the coverage holes. The proposed MAPC protocol works in two distinct phases.

Phase I aims at estimating the existing coverage provided by the randomly deployed static nodes. The widely used binary detection model, assumes that the sensing coverage of a sensor node is uniform in all directions, often represented by a unit disc. However, in reality, the sensing capabilities of a sensor are often affected by environmental factors. In particular, for range-based sensor modalities such as acoustics, radio, etc, the signal strength of the triggering signal decays as a function of distance. This implies that the detection capabilities of these sensors would exhibit similar characteristics as opposed to a uniform sensing range. In an effort to employ more realistic models in the computation of area coverage, in our work, we use the *Probabilistic Coverage Algorithm (PCA)* [8], that takes into account the probabilistic sensing behavior of sensor nodes.

Phase II manages relocation of the mobile sensors. For this phase, we propose a set of coverage and energy aware variants of *Virtual Force Algorithm*. These algorithms work in rounds and manage the mobile node relocation that serves a dual purpose. Firstly, this relocation *increases* the coverage during deployment by allowing the mobile nodes to fill in the coverage holes with minimal expenditure of energy. Secondly, the additional mobile sensors are uniformly spread in the target area for further discovery of coverage holes. The movements of the mobile nodes are controlled by novel thresholds based on real radio characteristics. This ensures that the nodes can communicate with each other, with high probability of successful transmission, during the round-by-round movements. The mobile nodes relocation in phase II thus results in increase in area coverage during the deployment stage. The individual phases of the proposed protocol are elaborated in detail in the subsequent sections.

4 Phase I: Coverage Estimation

We make the following assumptions:

- Static nodes are deployed in a random, non-deterministic fashion in an unknown obstacle-free environment.
- Location information is available using any existing GPS-less sensor network localization scheme.
- Sensors cannot detect the physical boundary of the region in outdoor environments. This assumption is consistent with the real world capabilities of existing sensor hardware.
- Nodes lying on the outer boundary of the deployed network topology (referred to as B-nodes) are aware of their special topological position using any of the existing outer boundary nodes identification schemes such as [9].

4.1 The Probabilistic Coverage Algorithm (PCA)

For coverage calculation, we use a realistic probabilistic coverage model proposed in our earlier work [8]. Using the probabilistic coverage model, the change in detection probabilities with distances can be represented by concentric circles drawn at constant distance increments around the sensor location. Each circle thus represents the probability of correctly receiving a signal, with strength above the receiving threshold, at a distance equal to the radius of that particular circle (see Figure 1). Assuming that the transmit power, P_t and receive threshold, γ, is known for a sensor through a priori field experiments and sensor calibration, a *probability table*, PT can be precomputed that provides the discrete detection probability at various distances from the sensor. If ρ_{reqd} represents the desired detection probability required by the application using the sensor network, the corresponding distance value, d_{reqd}, can be found from the PT.

For a deployed sensor network, a point in the target region can be covered by more than a single sensor. The overall detection probability Pr of a point in the region is thus given by Equation 1.

$$Pr = 1 - \prod_{i=1}^{N}(1 - Pr_i) \tag{1}$$

where N represents the number of sensor node covering a particular point and Pr_i denotes the detection probability of a point for a sensor i. The detection probability at any location is thus increased by contributions from all the sensors covering that point and it is possible to achieve the desired detection probability at distances greater than d_{reqd} from the sensor. The basic idea is to take the next higher distance from the probability table PT as d_{eval} (with lower detection probability than ρ_{reqd}) and evaluate whether contributions from neighbors makes the perimeter at d_{eval} sufficiently covered or not.

A node Sj that is a neighbor of Si has several concentric circles representing regions of different detection probabilities (see Fig 1). Node S_i calculates the cumulative detection probability at intersection of its circle at d_{eval} with various circles of neighbor S_j e.g., Pjs in Figure 1. $C(r, p)$ represent the circle around S_j with radius r providing probability of detection p.

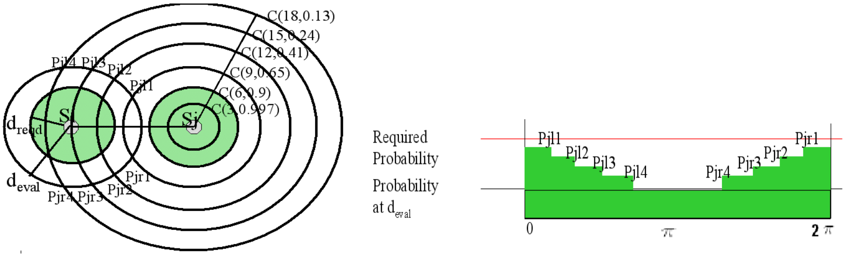


Fig. 1. Perimeter Coverage using PCA

The cumulative detection probability is then placed on a line segment $[0, 2\pi]$ representing the perimeter of S_i at d_{eval}. This is repeated for each neighbor until the whole perimeter is found covered by probability ρ greater than the required value. After executing the PCA, if a node finds that its evaluated perimeter is uncovered, the node can calculate a deployment location where a redundant helper node, S_h, can be placed such that the perimeter coverage constraint for the node is satisfied [8].

5 Phase II-Spreading Out of Mobile Sensors

Random aerial deployment is extremely challenging for mobile nodes as it may result in physical damage to their locomotive parts. In realistic deployments, the mobile nodes are normally accumulated at one or more points near the target area. We therefore, consider two different initial deployment methodologies namely *Normal* and *Island* distribution. In normal distribution, mobile sensors form a single cluster at the boundary, while in island distribution they form disconnected clusters at different locations on the boundary. For spreading of the clustered mobile sensor nodes, we propose to use the concept of virtual forces from robotics [1]. In this context, we propose two variants of *Virtual Force Algorithm (VFA)* (i) Coverage and Energy Aware VFA (CEA-VFA) and (ii) CEA-VFA with Simulated Movements (SM). For this phase, we assume that location information is available using any existing GPS-less WSN localization scheme for the mobile nodes such as [10] and that mobile nodes have significantly more initial energy than the static nodes.

5.1 Coverage and Energy Aware Virtual Force Algorithm (CEA-VFA)

We first provide a short overview of the basic VFA proposed in [1], [6]. Basic VFA attempts to iteratively spread the mobile sensors in the target area by using a combination of attractive and repulsive forces. Two mobile sensors will exert virtual forces on each other if the Euclidean distance, $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$, between them is not between a given range of thresholds, Th_{push} and Th_{pull} (discussed in detail later in this section). This virtual force, F_{ij} is a pull or attractive force if the distance between the two mobile nodes is greater than the pull threshold, Th_{pull}, while if the distance is less than the push threshold, Th_{push}, a push or repulsive force is exerted. Equation 2 shows the model used for decision making.

$$\boldsymbol{F}_{ij} = \begin{cases} F_{push}, & \text{if } d_{ij} < Th_{push} \\ 0, & \text{if } Th_{push} \le d_{ij} \le Th_{pull} \\ F_{pull}, & \text{if } d_{ij} > Th_{pull} \end{cases} \tag{2}$$

where $\boldsymbol{F}_{ij}$ is the force exerted on mobile node Si by neighbor Sj. Equations 3 and 4 represent the push/pull virtual forces.

$$F_{push} = (\frac{Th_{pull} + Th_{push}}{2}) - d_{ij} \tag{3}$$

$$F_{pull} = d_{ij} - (\frac{Th_{pull} + Th_{push}}{2}) \tag{4}$$

Representing magnitude of the force in terms of distance, the share of the virtual force for a node Si due to node Sj is represented by $\boldsymbol{dm}_{ij} = \frac{\boldsymbol{F}_{ij}}{2}$, the magnitude of which is denoted by $|dm_{ij}| = |(F_{ij}/2)|$. We express the total force exerted on a mobile sensor S_i by its n mobile neighbors, denoted by $\boldsymbol{F}_i$, as,

$$\boldsymbol{F}_i = \sum_{j=1, j \neq i}^{n} \frac{\boldsymbol{F}_{ij}}{2} = \sum_{j=1, j \neq i}^{n} \boldsymbol{d}m_{ij} \tag{5}$$

Note that $\boldsymbol{F}_i$ is the vector sum of all the forces acting on node S_i, the magnitude and orientation of which can be easily calculated, e.g., Robomote [11] uses an on-board compass combined with localization information for navigation purposes. The VFA works in rounds and the mobile nodes are iteratively moved to attain a more uniform distribution in the region.

Having explained the concept of virtual forces, we now detail the changes that have been adopted in CEA-VFA in the following sections.

Coverage Awareness. In CEA-VFA, a mobile node first checks whether it is in the vicinity of a coverage hole (has a neighbor static node with uncovered perimeter) and if so, it reacts to plug in the discovered hole before participating in the virtual force calculations. This coverage check is performed by each mobile node in each round of the CEA-VFA. Each round starts with mobile and boundary nodes (B-nodes) exchanging the location information using *Hello* messages. A Hello message contains the sender node ID, current location coordinates, and the remaining energy. A mobile node on reception of a Hello message marks the sender as its current neighbor and starts a TM_{wait} timer.

Static nodes with uncovered perimeter also start a TS_{wait} timer on receiving a Hello message from a mobile node. This timer is reset each time a Hello message is received. On expiry of the TS_{wait} timer, a static node with uncovered perimeter selects the nearest mobile node from its mobile node neighbor list and sends a *Help* message to the selected mobile node. The Help message contains sender node id, location, requested deployment point, and the expected gain in coverage (difference between the desired and existing detection probability).

A mobile node may receive multiple Help messages from different static nodes with uncovered perimeters. The mobile node selects the requested deployment point that involves the highest gain in coverage, broadcasts a *Move* message, and starts moving toward the requested deployment point. As the Move message is a broadcast, it is received by both mobile and static sensor nodes. Neighboring mobile nodes that receive the Move message, remove the sender node from the list of neighboring mobile nodes. The mobile nodes after expiry of the TM_{wait} timer perform the virtual force calculations to spread out in the topology to discover more coverage holes.

Energy Awareness. To ensure that the mobile nodes expend energy uniformly while in motion during the deployment phase, we introduce an adaptive policy based on the residual energy of the nodes. The virtual forces are made proportional to the residual energy of the node, with a higher energy node absorbing a greater portion of the virtual force than its low energy neighbor. Let E_I represent the initial energy of a mobile node, E_{ci} and E_{cj} represent the current remaining energies of nodes S_i and S_j respectively. We have $e_{ij} = (E_{ci} - E_{cj})/E_I$, where e_{ij} is the proportional energy coefficient. The magnitude of the force in terms of distance becomes $\frac{\boldsymbol{F}_{ij}}{2}(1 + e_{ij})$ instead of $\frac{\boldsymbol{F}_{ij}}{2}$.

Boundary Considerations. To keep the mobile sensors within the area bounded by B-nodes, B-nodes exert virtual forces ($\boldsymbol{F}_{ib}$) on mobile sensor nodes. A repulsive virtual force $\boldsymbol{F}_{ib}$ is included in the resultant vector sum of all virtual forces. The new total force is then expressed as,

$$\boldsymbol{F}_i = \sum_{j=1, j \neq i}^{n} \frac{\boldsymbol{F}_{ij}}{2}(1 + e_{ij}) + \sum_{b=1}^{k} \boldsymbol{F}_{ib} \tag{6}$$

where $\boldsymbol{F}_{ib}$ is the repulsive force exerted on the mobile node Si by its k neighbor B-nodes. If d_{ib} is the distance between Si and the B-node, $\boldsymbol{F}_{ib}$ is modelled by Equation 7.

$$\boldsymbol{F}_{ib} = \begin{cases} \frac{(Th_{push} + Th_{pull})}{2} - d_{ib}, & \text{if } d_{ib} < \frac{(Th_{push} + Th_{pull})}{2} \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

As B-nodes are all static nodes, mobiles nodes absorb all of the virtual force resulting from the B-nodes. As a final check, mobile nodes should not cross the virtual boundary formed by the known B-nodes.

Choice of Thresholds, $\boldsymbol{Th_{push}}$ and $\boldsymbol{Th_{pull}}$. We want to ensure that the mobile nodes are able to communicate with neighbors during the round by round operation of the algorithm. Rather than using static values of movement triggering thresholds, Th_{push} and Th_{pull}, these thresholds should depend on the link quality. Sensor nodes can be placed further apart (higher value of Th_{pull} can be used) for good quality communication links. Although other complex models can also be used, we use a simple radio propagation model based on log-normal shadowing [12] to characterize the communication link between two sensors.

Each sensor has a *receive threshold* value γ that describes the minimum signal strength that can be correctly decoded at the sensor. The probability Pr that a received signal level, P_{rec} at a sensor will be above this receive threshold, γ,

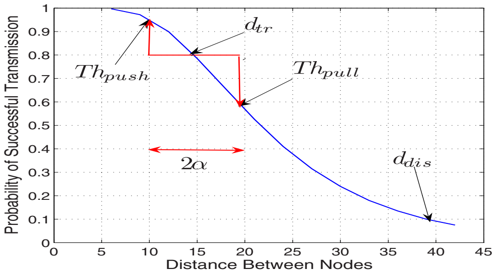


Fig. 2. Change in communication probability with distance

is given by Equation 9, with Q-function to compute probability involving the Gaussian process. The Q-function is defined as

$$Q(z) = \frac{1}{\sqrt{2\pi}} \int_{z}^{\infty} exp(-\frac{x^2}{2})dx \tag{8}$$

where $Q(z) = 1 - Q(-z)$.

$$Pr[P_{rec}(d) > \gamma] = Q[\frac{\gamma - P_{rec}(d)}{\sigma}] \tag{9}$$

For a given transmit power and receive threshold value, we can calculate the probability of receiving a signal above the receive threshold value, γ, at a given distance using Equations 8 and 9 as shown in Figure 2.

Following the terminology in [13], there are three distinct reception regions in a wireless link: connected, transitional, and disconnected. The transitional region has highly unreliable links and its region bounds can be found either by analytical or empirical methods [13]. Let d_{tr} and d_{dis} represent the points where the transitional and disconnected regions begin respectively. We define Th_{push} and Th_{pull} as $(1 - \alpha)d_{tr}$ and $(1 + \alpha)d_{tr}$ respectively, where α denotes the error tolerance coefficient. Note that the values of Th_{push} and Th_{pull} are bounded by d_{dis}. α reflects the tolerance to the errors in localization and odometry during navigation of the mobile nodes. As long as the final position after movement is within this range, the deviation from the ideal trajectory during movement can be tolerated by our movement algorithm. Figure 2 shows d_{tr}, Th_{push}, Th_{pull} and d_{dis} in terms of probability of correct packet reception.

5.2 Virtual Force Algorithm with Simulated Movement

Mobile nodes move after each round of CEA-VFA due to the virtual force exerted by their neighbors before settling down to their final position in the topology. If we could calculate the final position of a mobile node and move directly to that final position, we can save energy by moving much lesser distances, i.e., by using the cut-through paths instead of the zig-zag paths that result from the round-by-round operation of CEA-VFA.

We propose CEA-VFA with Simulated Movement (SM) approach that attempts to use the cut-through paths for movement. Nodes go through the CEA-VFA iterations and calculate new position after each round. The difference is that nodes do not physically move after each iteration, rather, they stay at their original position and simply assume the new calculated virtual position. The Hello messages at the start of the next round contain the new virtual positions enabling the recipients to use this updated position information for the upcoming round of CEA-VFA. Nodes only move once the algorithm terminates.

This simplistic approach has a few disadvantages. B-nodes, coverage holes, and new mobile neighbors are discovered by the per round movements in CEA-VFA. If a fully simulated run of CEA-VFA is used, it is highly likely that the mobile nodes may not detect the presence of the other mobile nodes and B-nodes in the region. Similarly, it is not possible to locate all of the coverage holes. One way to offset this disadvantage is to use intermittent simulated movement instead of fully simulated movement. In the CEA-VFA Intermittent Simulated Movement (ISM) approach, nodes simulate the movement for x number of rounds. Actual physical movement only takes place after every x number of rounds e.g., in ISM2 ($x = 2$) nodes physically move after every second round. Similarly, in ISM6 a node physically moves after every sixth round of virtual force calculations.

The simulated movement approach does not incur any additional communication overhead than the CEA-VFA rather it reduces energy consumption due to movement by moving the mobile sensors in fewer number of rounds. This also results in quicker deployment due to reduction in the time spent in the per round zigzag movements of the CEA-VFA.

5.3 Performance Evaluation

We conducted NS2 simulations for basic VFA, CEA-VFA and different variants of ISM and compared the results with an centralized optimized assignment.

A Centralized Optimization. We want to optimally assign the available mobile sensors, firstly, to locations requested by static nodes and secondly, to locations in the topology so as to form a uniform distribution. This is a classical *Assignment* optimization problem, also referred to as *bipartite weighted matching* problem in graph theory. A centralized optimization can be performed, if we assume that the location information of all coverage holes and the existing positions of all mobile nodes are known.

The assignment problem can be formulated as following: Let S_m represent the set of p mobile nodes, $S_m = \{S_{m1}, S_{m2}, \ldots, S_{mp}\}$ and set S_h represent the location of q coverage holes in the topology, $S_h = \{S_{h1}, S_{h2}, \ldots, S_{hq}\}$. After assigning q out of p mobile nodes, we have r remaining mobile sensors, ($r = p - q$), that needs to be uniformly distributed in the coverage area. The set S_u represent the deployment locations of these r mobile nodes, $S_u = \{S_{u1}, S_{u2}, \ldots, S_{ur}\}$. Let $S_d = \{S_{d1}, S_{d2}, \ldots, S_{dt}\}$ represent the combined deployment locations with $S_d = S_h \cup S_u$ and $t = q + r$. If a mobile node i is assigned to a location j, there is a cost (energy consumption due to movement) of c_{ij}. We can minimize the total value of assignment (minimize energy consumption). The objective function can be defined as:

"What is an assignment schedule in order to minimize energy consumption?"

There are additional constraints that each mobile node can be assigned to exactly one location and each location must have one assigned mobile node. The assignment problem can be formulated as an Integer Linear Program (ILP) as follows;

$$Minimize \sum_{j=1}^{t} \sum_{i=1}^{p} c_{ij} x_{ij}$$

Constraints:

- $x_{ij} \leq 1; i = 1, \ldots, p; j = 1, \ldots, t$
- $\sum x_{ij} \leq 1; j = 1, \ldots, t$
- $\sum x_{ij} \leq 1; i = 1, \ldots, p$

Simulations Results. We implemented the B-node selection algorithm [9], PCA, basic VFA, CEA-VFA and simulated variants in discrete event simulator NS2. Static nodes are randomly deployed in a 100m by 100m region with value of ρ_{reqd} as 0.9. Values of Th_{push}, Th_{pull}, and d_{dis} were set at 25m, 33m, and 40m respectively. Initial energy of each node was 4528J (3.7v, 345mAh) and energy consumed in movement was taken as 8.274 J/m (Robomote [11]). In island distribution, mobile nodes form two disconnected clusters at opposite corners of the topology. Maximum number of rounds was set to 12. The results are averaged over three different topologies for each type of deployment.

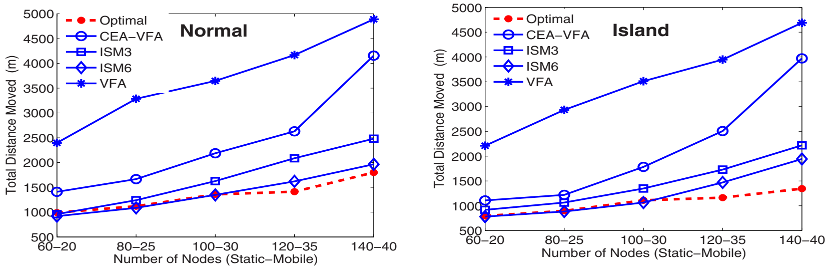


Fig. 3. Total distance moved

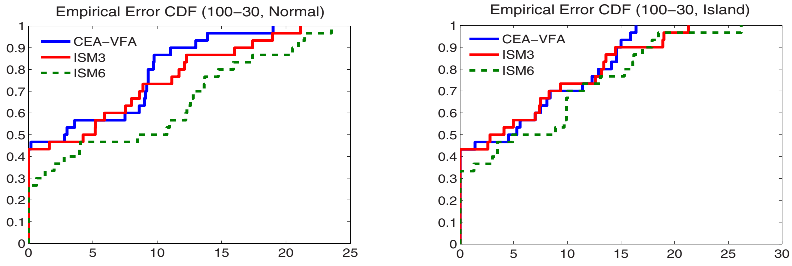


Fig. 4. Empirical error CDF for Normal and Island distributions

Figure 3 compares the total distance moved by all mobile nodes. The results show that CEA-VFA and the simulated variants perform better than the basic VFA, for both normal and island initial deployment, for different numbers of mobile nodes. On average, CEA-VFA causes the mobile nodes to move about 63% and 57% of the total distance moved in case of basic VFA for normal and island deployments respectively. This saving is primarily due to the coverage awareness that enables CEA-VFA to discover and plug the coverage holes in each round of the VFA. Comparing CEA-VFA with the simulated variants, CEA-VFA causes the mobile sensors to move the highest total distance for all types of topologies and initial deployments. On the other hand, mobile nodes using ISM6 variant consistently moves the least distance. This is because in ISM6, the mobile sensors only moves in two of the twelve rounds. Also note that in some cases, mobile nodes using ISM6 moves a lesser total distance than that given by optimized assignment (e.g., 60-20, 80-25, and 100-30 nodes topologies, for both normal and island initial deployments). This is because in these cases ISM6 not only fails to discover all of the coverage holes in the topology but it also produces a poor topology distribution by moving lesser distances.

Figure 4 illustrates the empirical error Cumulative Distribution Function (CDF) for both normal and island distributions for 100 static-30 mobile nodes topologies. Errors are calculated as the difference between the desired deployment points and the final topology position achieved by different virtual force based algorithms. A zero error means that either a coverage hole has been plugged or a perfect grid point deployment has been achieved. For island distribution, the error CDF of ISM3 matches closely with that of CEA-VFA while ISM3 moves about 55% of the total distance moved by CEA-VFA. For ISM6, lesser number of nodes (about 27-33%) report zero deployment error than ISM3 and CEA-VFA (about 43-47%). Also the spread in error CDF is more for ISM6 than either ISM3 and CEA-VFA.

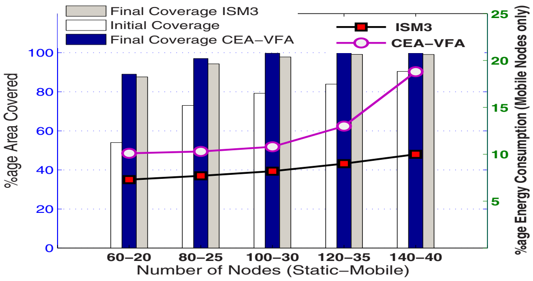


Fig. 5. Percentage of area covered and energy consumed

Figure 5 shows the initial and final percentage of area with sufficient coverage (shown by bars) for CEA-VFA and ISM3 with different topologies. Mobile nodes spread out from an initial island distribution. For 100-30, 120-35, and 140-40 static-mobile node cases, more than 99% of the area is covered after relocation of mobile nodes for both CEA-VFA and ISM3. For 80-25 configuration, coverage is enhanced from 72% to 94% after execution of 12 rounds of the ISM3 algorithm while the corresponding coverage in CEA-VFA is 96.7%. This gain in coverage is at the expense of energy consumption due to relocation of the mobile nodes. Figure 5 also shows that for 140-40 topology, mobile nodes using ISM3 only consume about 10% of their total initial energy in the deployment phase as compared to close to 19% for CEA-VFA (shown by lines).

Simulation results show that ISM variants save a considerable amount of energy by moving the mobile nodes lesser distances than the VFA and CEA-VFA. However, this saving is at the cost of slight non-uniformity in the node distribution. Performance of ISM3 is comparable to ISM6 in terms of energy consumption and yet it achieves coverage closer to that of CEA-VFA. To summarize, the simulation results show that ISM3 is a good compromise with significant savings in energy consumption.

6 Conclusion

In this paper, we have proposed a distributed protocol MAPC, for providing adequate coverage of the target area using a combination of mobile and static sensor nodes. Most of the assumptions made in our protocol are realistic and implementable in real-life aplications. Our discrete event simulation results demonstrated that, for different type of initial deployments, our protocol consumes only 30-40% of the energy consumed by the basic virtual force algorithm. MAPC is thus successful in enhancing area coverage to the desired degree while ensuring that minimal energy is expended by the mobile nodes.

In future, we plan to extend the protocol to incorporate obstacles during coverage calculation and the mobile node relocation phases. We also plan to carry out experiments to validate the working of the proposed protocol.

References

1. Howard, A., Mataric, M.J, Sukhatme, G.S: Mobile sensor network deployment using potential fields:A distributed, scalable solution to the area coverage problem. In: DARS 2002. 6th International Symposium on Distributed Autonomous Robotics Systems (June 2002)
2. Wang, G., Cao, G., Porta, T.L.: A bidding protocol for deploying mobile sensors. In: ICNP 2003. 11th IEEE International Conference on Network Protocol, pp. 315–324 (November 2003)
3. Wang, G., Cao, G., Porta, T.L.: Movement-assisted sensor deployment. In: IEEE INFOCOM 2004 (June 2004)
4. Howard, A., Mataric, M.J, Sukhatme, G.S: An incremental self-deployment algorithm for mobile sensor networks. Autonomous Robots, Special Issue on Intelligent Embedded Systems 13(2), 113–126 (2002)
5. Batalin, M.A., Sukhtame, G.S.: Coverage, exploration and deployment by a mobile robot and communication network. Telecommunication Systems Journal, Special Issue on Wireless Sensor Networks 26(2), 181–196 (2004)
6. Zou, Y., Chakrabarty, K.: Sensor deployment and target localization in distributed sensor networks. ACM Transactions on Embedded Computing Systems 2(3), 1–29 (2003)
7. Wang, G., Cao, G., Porta, T.L.: Proxy based sensor deployment in mobile sensor networks. In: MASS 2004, pp. 493–502 (October 2004)

8. Ahmed, N., Kanhere, S.S., Jha, S.: Probabilistic coverage in wireless sensor networks. In: LCN 2005, pp. 672–681 (November 2005)
9. Ahmed, N., Kanhere, S.S., Jha, S.: Efficient boundary estimation for practical deployment of mobile sensors in hybrid sensor networks. In: MASS 2006, pp. 662–667 (October 2006)
10. Moore, D., Leonard, J., Rus, D., Teller, S.: Robust distributed network localization with noisy range measurements. In: Proceedings of the ACM SenSys 2004 (2004)
11. Dantu, K., Rahimi, M., Shah, H., Babel, S., Dhariwal, A., Sukhatme, G.: Robomote: Enabling mobility in sensor networks. Technical Report CRES-04-006, University of Southern California (2004)
12. Rappaport, T.S.: Wireless communications: Principles and Practice. Prentice Hall, New Jersey (1996)
13. Zuniga, M., Krishnamachari, B.: Analyzing the transitional region in low power wireless links. In: SECON 2004 (October 2004)

Area Localization Algorithm for Mobile Nodes in Wireless Sensor Networks Based on Support Vector Machines

Bin Yang, Jianhong Yang, Jinwu Xu, and Debin Yang

School of Mechanical Engineering, University of Science and Technology Beijing, Beijing, 100083, China
jpuyang007@gmail.com

Abstract. Many applications in wireless sensor networks require sensor nodes to obtain their absolute or relative positions. Although various localization algorithms have been proposed recently, most of them require nodes to be equipped with range measurement hardware to obtain distance information. In this paper, an area localization method based on Support Vector Machines (SVM) for mobile nodes in wireless sensor networks is presented. Area localization is introduced as an evaluation metric. The area localization procedure contains two phases. Firstly, the RF-based method is used to determine whether the nodes have moved, which only utilizes the value change of RSSI value rather than range measurement. Secondly, connectivity information and SVM algorithm are used for area localization of mobile nodes. The area localization is introduced to trade off the accuracy and precision. And area localization, as a new metric, is used to evaluate our method. The simulation experiments achieve good results.

Keywords: Area localization; Mobile nodes; Support Vector Machines; Wireless sensor networks.

1 Introduction

Wireless sensor network is a distributed collection of nodes which are resource constrained and capable of operating with minimal user attendance. Wireless sensor nodes operate in a cooperative and distributed manner. Such nodes are usually embedded in the physical environment and report sensed data to a central base station.

Many applications of wireless sensor networks, including condition monitoring of the mechanical equipments, environmental monitoring, military surveillance, search-and-rescue operations and other commercial applications, need the approximately geographic positions of nodes, especially obtain the positions of mobile nodes. Localization is an inevitable challenge when dealing with wireless sensor nodes, and a problem which has been studied for many years. Nodes can be equipped with a Global Positioning System(GPS)[1], but it is costly solution and power consumption.

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 561–571, 2007.

While many studies have focused on developing different algorithms for localization, less attention has been paid to the problem of area localization. For most purposes accurate point in localization is rarely necessary. For example, condition monitoring of the equipments, especially large equipments, no point exact detection is necessary but rather range detection. An action should be triggered when entering leaving or switching areas. The idea is to sacrifice precision in terms of being at an exact location for accuracy, being at a possible set of location. This tradeoff would create a more robust system, less susceptible to environmental changes.

Almost hundreds of nodes whose locations are known would be placed to sense the physical environment in many applications. The new locations of the objects should be acquired when they have been moved. Huge cost will be paid to access the precise coordinates of the nodes with little significance. Because of limitations to the existing localization algorithm, we need to develop new localization methods and evaluation metrics, such as area localization.

1.1 Related Work

There are many dimensions to categorize existing techniques, such as centralized vs. decentralized, beacons vs. beacon-less, and ranging vs. ranging-free.

Centralized localization techniques depend on sensor nodes transmitting data to central location, where computation is performed to determine the location of each node. Doherty, Pister and Ghaoui develop a centralized technique by using convex optimization to estimate positions based only on connectivity constraints[2]. MDS-MAP[3] improves on these results by using a multidimensional scaling approach.

Distributed localization methods do not require centralized computation, and rely on each node determining its location with only limited communication with nearby nodes. These methods can be classified as the range-based and the range-free. Range-based techniques use distance estimates or angle estimates in location calculations, while a range-free solution only depends on the contents of the received messages.

Range-based approaches have exploited time of arrival (TOA)[4], received signal strength [5], time difference of arrival of two different signals (TDOA) [6], and angle of arrival (AOA) [7~8]. Though they can reach fine resolution, either the required hardware is expensive (ultrasound device for TDOA, antenna arrays for AOA) or the results depend on other unrealistic assumptions about signal propagation (for example, the actual received signal strengths of radio signals can vary when the surrounding environment changes).

Because of the hardware limitations of sensor devices, range-free localization algorithms are cost effective alternatives to more expensive range-based approaches. There are two main types of range-free localization algorithms proposed for sensor networks: local techniques that rely on high density of anchors nodes so that every node can hear several anchors nodes, for example the APIT method [9]; and hop counting techniques that rely on flooding a network , for example DV-HOP [10] and the Amorphous localization algorithm [11].

Recently, Duc A develops Support Vector Classification Strategies for Localization in Sensor Networks[12]. But this method does not consider the localization of mobile nodes and pays costly in order to obtain coordinator of unknown nodes. In this paper, we will extend this idea by using SVM to area localization for mobile objects.

1.2 Our Contribution

Compared the existing localization methods in wireless sensor networks, the coordinate of nodes is calculated by most of them, but the precision is not satisfactory. Generally, these methods have not solved the problem how to determine the positions of mobile nodes.

In our method, we make full use of RF resource, and greatly reduce the impact of RF instability. The RF method is utilized to determine whether the nodes have moved, but it does not need range measurement, only need to know the change of RSSI value. And an average method is used to reduce the RF instability by environment changes. On the other hand, area localization is introduced in our methods. Connectivity information of nodes and SVM method are made full use of to return the area of mobile nodes. In simulated experiments we utilize new assessment metrics to evaluate our methods.

1.3 Paper Organization

The rest of this paper is organized as follows: Section 2 introduces the relative definitions and terms. Section 3 provides the methods to determine nodes on mobile objects whether they have been moved. Section 4 provides a method based on SVM to area localization for mobile nodes. Performance evaluation is presented in Section 5. Finally, Section 6 concludes the whole paper.

2 Terms and Definitions

There are n anchor nodes(AN) on a field, AN_1, AN_2,…AN_n. The offline measured signal strength is called the training set. A training set consists of a set of RSSI value of per anchor node. Every node will collect the signal strength reading from anchor nodes which can directly communicate, T_0={ $RSSI_i$}, i = 1,…m, where m is the total number of the anchor nodes in communication range. All information will be used to describe the mobile nodes.

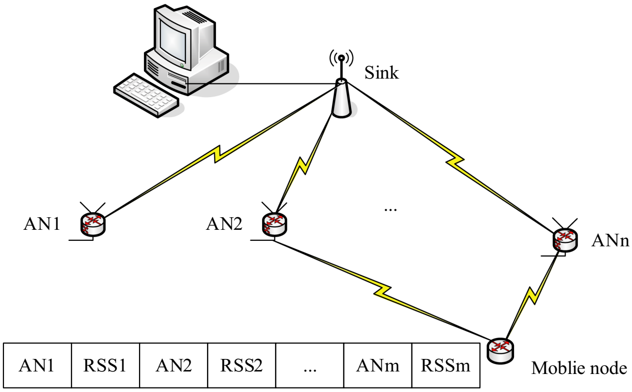
In this paper, a novel area-based presentation approach for WSN mobile nodes localization systems is introduced. The purpose of the area-based localization system is to return the possible locations of the mobile objects as an area rather than a single location. The important property that area-based systems exhibit in dealing with the uncertainty is their ability to trade off accuracy for precision.

For area-based systems, we define accuracy as localization error of the returned area. Precision describes the size of the area.

As is mentioned above, the higher the accuracy is, the more area are returned. This increases the accuracy but the overall precision decreases. So our method will trade off accuracy and precision.

3 Determination of Mobile Nodes

In this section Mobile Confirm System Architecture (MCSA) is presented to determine whether the nodes have been moved (Figure 1). As any other radio frequency estimation system, MCSA needs to monitor received signal strength from anchor nodes. In practical application, according to the field area and communication range of anchor nodes, suitable number of anchor nodes can be chosen. Every node must be able to communicate with at least one other anchor node at anytime.



AN1	RSS1	AN2	RSS2	...	ANm	RSSm

Fig. 1. Mobile Confirm System Architecture. Each mobile node stores the RF value which comes from the anchor nodes. According to the average value of RF value, the mobile node determines whether it has been moved.

An efficient and simple mechanism is implemented to decide whether nodes have been moved or not. Figure 2 shows this mechanism. Server will receive notification when the nodes receive consecutive different values of signal strength. Since the RSS is not stable and changes every second, the average value of last three RSS samples from each anchor node ($RSS_{avg\text{-}i}$) is calculated. According to system requirements, we define two threshold, thresholdN and thresholdM. Obviously, the thresholdN describes the minimum change of RSS value. When the place has changed, the received signal strength changes every second. Therefore the conclusion may be incorrect if the signal strength variation is too significant. To solve this problem, a simple but efficiently strategy is proposed, that is, the max threshold (thresholdM, in Fig 2) is defined.

These latest average values are compared with the last average value($RSS_{lastavg-i}$) which stored in the node. If the absolute value of one among these differences is greater than the thresholdN or lesser than the thresholdM, it indicates the nodes have been moved. Otherwise, we consider the nodes don't move obviously or are static. Then the mobile nodes will send a notification to sink, server will know the nodes have been moved. These thresholds including thresholdN and thresholdM will be decided by user according to the actual environment. On the other hand, if all differences are smaller than thresholdN or greater than thresholdM, MCSA supposes the node is static and no notification is sent.

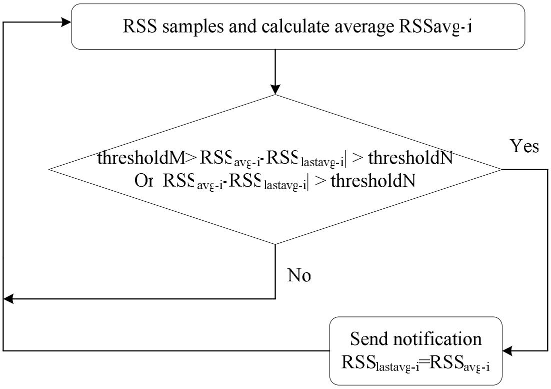

RSS samples and calculate average RSSavg-i
thresholdM> RSSavg-i-RSSlastavg-i| > thresholdN
Or RSSavg-i-RSSlastavg-i| > thresholdN
Yes
No
Send notification
RSSlastavg-i=RSSavg-i

Fig. 2. Mobile nodes determine flowchart. The MCSA compares the difference of the adjacent RSSI average value with thresholdN and thresholdM. If it meets the conditions, the mobile nodes will sent the message to sink node and trigger the localization process.

Generally speaking, the thresholdN is more important than thresholdM to evaluate the mobile nodes. In area localization, how can the thresholdN be obtained?

To evaluate the thresholdN of RSSI, the nodes which we developed equipped with Atmel128 as MCU and CC1100 as RF chip. We assume the area is square which side length is L in our experiments. Then the approximate thresholdN is computed as:

$$\text{thresholdN} = 10n\log(L/d_0)$$

where
n : path loss exponent
d_0: the close-in reference distance
L: the side length of the square field

Obviously, we also can use other manner to evaluate thresholdN such as statistical methods.

Finally, the node will update the memory where the latest RSS_{avg-i} value is stored. When MCSA knows that a number of nodes have been moved, it will trigger the area localization by using SVM method.

4 Mobile Nodes of Area Localization Based on SVM Strategy

4.1 Network Model

A large scale wireless sensor network of N nodes $\{S_1, S_2, \ldots, S_N\}$ are deployed in a 2-d geographic area $[0, D] \times [0, D]$ $(D > 0)$. According to practical conditions, this field is divided to $M_1 \times M_2$ area where M_1 and M_2 should be depended on the precision and accuracy of area localization. Generally, we assume that M_1 equals to M_2. Anchor nodes don't participate in the area localization, and only contribute to mobile nodes confirmation. $r(S_i)$ denotes the communication range of each node S_i , and this paper assumes every node has the same r $(r > 0)$. Two nodes are said to be "reachable" from each other if there exists a path of communication between them. k beacon nodes which come from N nodes are chosen for every once area localization. And the k beacon nodes know their own location and are reachable from each other. When nodes have changed their place, a new area localization algorithm of mobile nodes is devised. Mere the connectivity of the networks needs to be known.

4.2 SVM Model and Classification Strategy

Let $(x(S_i), y(S_i))$ denote the true (to be found) coordinates of node S_i's location, and $h(S_i, S_j)$ depict the hop-count length of the shortest path between nodes S_i and S_j . Each node S_i is represented by a vector $s_i = <h(S_i, S_1), h(S_i, S_2), ..., h(S_i, S_k)>$. The training data for SVM is the set of beacons $\{S_i\}$ $(i = 1 \sim k)$. A Radial Basis Function is defined as the kernel function because of its empirical effectiveness .

The implementation for SVM strategies has two kinds multi-class strategy (MCS) and the decision-tree strategy (DTS). Multi-class classification includes One-Against-All, One-Against-One, and DAGSVM[13]. In order to decrease the computation cost, MCS classification strategy is adopted in this paper.

They are presented in the following sections.

We consider sets of M_1 and M_2 classes to classify nodes.

- M_1 classes for the x dimension $\{cx_1, cx_2, \ldots, cx_{M1}\}$. Each class cx_i contains nodes with $x >= i \times D/M_1$.
- M_2 classes for the y dimension $\{cy_1, cy_2, \ldots, cy_{M2}\}$. Each class cy_i contains nodes with $y >= i \times D/M_2$.

Intuitively, each x-class cx_i contains nodes that lie to the right of the vertical line $x = i \times D/M_1$, while y-class cy_i contains nodes that lie above the horizontal line $y = i \times D/M_2$. Therefore, if the SVM learning predicts that a node S is in class cx_i but not class cx_{i+1}, and in class cy_j but not class cy_{j+1}, we conclude that S is inside the square cell $[i \times D/M_1, (i +1) \times D=M_1] \times [j \times D/M_2, (j + 1) \times D/M_2]$. The square cell is the area localization of mobile nodes.

4.3 Algorithms

As is mentioned above, when the mobile nodes change their places, area localization is triggered. Area localization of each mobile node can be estimated using the following algorithm. Fig.3 shows the main flowchart used for the simulation experiments.

Algorithm (2-d Area Localization)

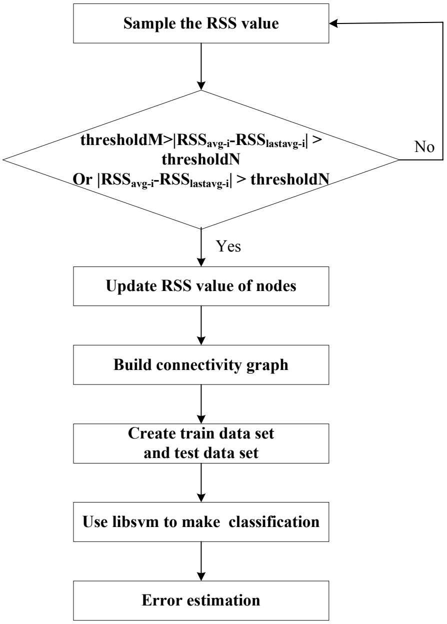

Sample the RSS value
thresholdM>|RSS_{avg-i}-$RSS_{lastavg-i}$| > thresholdN
Or |RSS_{avg-i}-$RSS_{lastavg-i}$| > thresholdN
No
Yes
Update RSS value of nodes
Build connectivity graph
Create train data set and test data set
Use libsvm to make classification
Error estimation

Fig. 3. The algorithm flowchart. If the mobile nodes determine it has been moved, it will updates the RSSI value in the memory. Then the connectivity graph will be built once again. Subsequently, the localization algorithm based on SVM will be run.

Step 1. Calculate the average value of three times sample RSS reading value. Compare the latest RSS value with the last RSS value stored in nodes, if the difference value is larger than the threshold value thresholdN and less than the threshold value thresholdM, the area localization will be triggered.

Step 2. SVM for area localization

At the beginning, the connectivity graph is built, including the beacon nodes and mobile nodes. The information helps build hops matrix. And the hop count forms support vector. Then localization accuracy will be defined. According to the localization accuracy, the X axis and Y axis are divided. Finally, libsvm is used to estimate area localization of mobile node.

Step 3. Error estimation

After step 2, the field of mobile nodes has been obtained. Then we can compare the estimation value with real value. Finally, the average area localization error, maximum area localization error, standard deviation area localization error will be obtained.

5 Performance Evaluation

In this section the performance of metrics are described. These metrics are used to evaluate the efficiency of our methods. The traditional localization metric is the distance error between the returned position and the true position. There are many ways to describe the distribution of the distance error, for example the average. The problem of traditional metric is that it is not proper for area-based approaches. Thus we introduce new metrics which are appropriate for area-based algorithm.

Distance Accuracy. This metric is the distance between the true area center and area center of the returned area.

Precision. The overall precision refers to the size of the returned area, i.e. the m^2. To normalize the metric across environments, it can be expressed as a percentage of the entire possible space (i.e. the size of the plant).

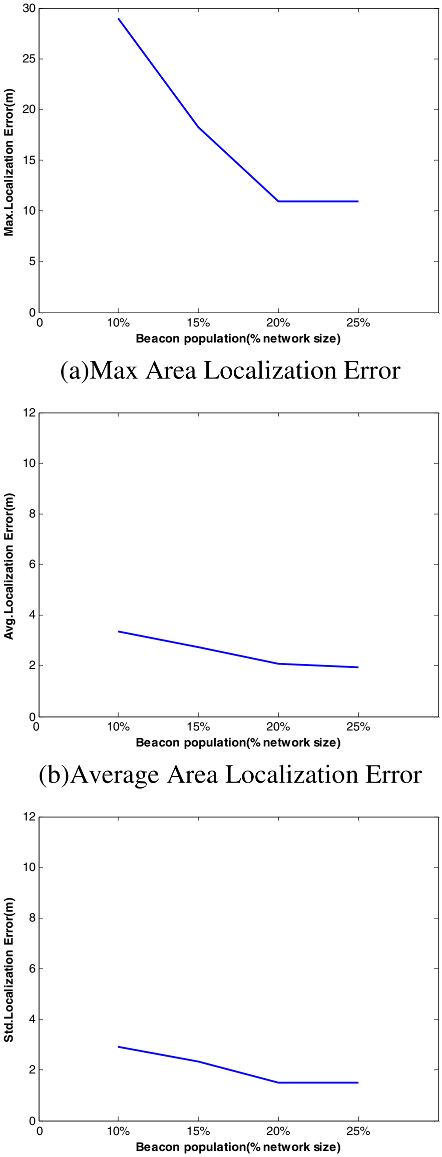
In practice, we divide a plant into a set of fields where each field is a rectangular or square area.

In our performance evaluation, only the area localization procedure is considered. We assume that the first step has finished, and the mobile nodes have be found.

We conducted a simulation study on a network of 1000 sensors located in a 100m × 100m 2-D area. We assumed uniform random distribution for the sensor locations and the selection of the beacon sensors. We considered the communication range of all the nodes is 10m. We also considered four different beacon populations: 10% (k = 100 beacons), 15% (k = 150 beacons), 20% (k = 200 beacons), and 25% (k = 250 beacons). The communication cost is mainly due to k broadcasts where k is the number of beacons. We, therefore, focus more on the localization accuracy.

We used the algorithms in the libsvm[14] software for SVM classification. In the following sections, we discuss the results (in present tense for each of presentation) of the following studies: quality and efficiency of SVM classification, comparisons of our method in different classification numbers (e.g. different in M), under the effects of beacon population, network density and so on.

According to the practical environment, the designed precision is 1m. How can we get this goal? In order to achieve this goal, firstly, we assume the M equals 127. The precision is 100/128(0.78125m), which meets our requirements. Secondly, we assume the M equal 63, the precision is 100/64(1.5625m). Obviously, the value(1.5625m) is larger than 1m, it does not meet our requirement. So we consider the M equals 127 in our simulation experiments. The results of experiments are as follows (Fig 4):



(a)Max Area Localization Error

(b)Average Area Localization Error

(c)Standard Deviation

Fig. 4. Multi-Class (communication range 10m): Statistics on area localization error of each technique under different beacon populations

Figure 4 plots the average, max, and standard deviation of area localization errors for the range-10m network. It is understandable that when this scale of beacon nodes is larger, the area localization is more accurate. The simulation results perform well when 25% of the network serve are beacons, in which case the max area localization error is no more than 12m. The average area localization is about 2m. The standard deviation is also small (about 2m), indicating that most sensors' locations are very well estimated. It is understandable that SVM's accuracy increases with the beacon population, This experiments result strongly supports our approach of using SVM classification for the sensor area localization problem.

The remainder of this section, the border problem is discussed. Sensor nodes close to the edge of the sensor field are poorly positioned compared to those deep inside the field. We use the four kinds beacon population to investigate this problem in simulation experiments. In Fig.6(a), the coordinator of the sensor node which has the max error is closed to the edge. We will research border problem in the future.

6 Conclusion

An approach about area localization of mobile nodes based on the SVM is presented in a large-scale sensor networks. In this paper the concept of area localization is introduced to trade off accuracy and precision, and two new evaluation metrics are provided to access the area localization. In our methods, the value change of RSS, connectivity information and SVM methods are used. In our SVM model, the beacon nodes serve as the training points and the feature vector uses only connectivity information. Therefore, the proposed approach can avoid instability of RF based method. And the SVM method does not require expensive ranging and specialized devices.

Our simulation study has shown that the approach provide good area localization accuracy. Our future research includes a comparison our method with other existing localization techniques as well as alleviate the compute cost to apply it in practice.

Acknowledgments. This research project is supported by National Natural Science Foundation of China (under Grant No. 50674010). And this project is to support the work and recommendations of the students, in particular, School of Mechanical Engineering, University of Science and Technology of Beijing Dr. Limin, Dr. Liu Feng, Dr. Sun Yuanbo.

References

1. Hofmann, W.B., Lichtenegger, H., Collins, J.: Global Positioning System: Theory and Practice, 2nd edn. Springer, New York (1993)
2. Lance, D., Kristofer, P., Laurent, E.G.: Convex Position Estimation in wireless Sensor Networks. In: Proceedings of IEEE INFOCOM, pp. 1655–1663. IEEE, USA (2001)
3. Paramvir, B., Venkata, N.: RADAR: An In-Building RF-Based User Location and Tracking System. In: IEEE InfoCom 2000, vol. 2, pp. 775–784 (2000)
4. Harter, A., Hopper, A., Steggles, P.: The anatomy of a context-aware application. In: Proceedings of MOBICOM 1999, Seattle, Washington, pp. 59–68 (1999)

5. Girod, L., Bychovskiy, V., Elson, J., Estrin, D.: Locating tiny sensors in time and space: A case study. In: Werner, B. (ed.) Proceeding of 2002 IEEE International Conference on Comoyter Design: VLSI in Computers and Processors, pp. 214–219. IEEE Computer Society, Freiburg, Germany (2002)
6. Girod, L., Estrin, D.: Robust range estimation using acoustic and multimodal sensing. In: IROS 2001. Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 3, pp. 1312–1320. IEEE Robotics and Automation Society, Maui (2001)
7. Priyantha, N.B., Chakraborty, A., Balakrishnan, H.: The cricket location-support system. In: Proceedings of the 6th ACM MOBICOM, pp. 32–43. ACM, New York, USA (2000)
8. Niculescu, D., Nath, B.: Ad hoc positioning system (APS) using AoA. In: Proceedings of IEEE INFOCOM 2003, pp. 1734–1743 (2003)
9. He, T., Huang, C., Blum, B.M., Stankovic, J.A.: Range-free localization schemes in large scale sensor networks. In: Proceedings of the Ninth Annual International Conference on Mobile Computing and Networking, pp. 81–95 (2003)
10. Niculescu, D., Nath, B.: Dv based positioning in ad hoc networks. J. Journal of Telecommunication Systems 22, 267–280 (2003)
11. Nagpal, R., Shrobe, H., Bachrach, J.: Organizing a global coordinate system from local information on an ad hoc sensor network. In: Zhao, F., Guibas, L.J. (eds.) IPSN 2003. LNCS, vol. 2634, pp. 333–348. Springer, Heidelberg (2003)
12. Duc, A., Nguyen, T.: Localization in Wireless Sensor Networks based on Support Vector Machines. Submitted to IEEE Transactions on Parallel and Distributed Systems (2007)
13. Hsu, C.W., Lin, C.J.: A comparison of methods for multi-class Support Vector Machines. IEEE Transactions on Neural Networks 13, 415–425 (2002)
14. Chang, C., Lin, C.: LIBSVM – A library for Support Vector Machines, National Taiwan University, http://www.csie.ntu.edu.tw/cjlin/libsvm

A Dual-Token-Based Fault Tolerant Mutual Exclusion Algorithm for MANETs

Weigang Wu[1], Jiannong Cao[1], and Michel Raynal[2]

[1] Department of Computing, The Hong Kong Polytechnic University,
Kowloon, Hong Kong
{cswgwu,csjcao}@comp.polyu.edu.hk
[2] IRISA, Campus de Beaulieu, Université de Rennes1,
35042 Rennes Cedex, France
raynal@irisa.fr

Abstract. Most existing mutual exclusion algorithms for mobile ad hoc networks (MANETs) adopt a token-based approach. In traditional wired networks, timeout-based mechanisms are commonly used to detect token losses. However, in MANETs, it is difficult to set a proper timeout value due to the network dynamics. In this paper, we propose a dual-token-based mutual exclusion algorithm, which can tolerate token losses without using timeout. Two tokens are concurrently circulated in the system to monitor each other by using sequence numbers. If one token is lost, the other token can detect the loss and regenerate a new token. Simulations have been carried out to evaluate the effectiveness and performance of the proposed algorithm in comparison with the timeout-based approach. The results show that the timeout-based algorithm may falsely claim the loss of a token, thus cannot guarantee the correctness of mutual exclusion algorithms. On the contrary, our proposed algorithm can avoid false detection of token losses and satisfy all the correctness requirements of mutual exclusion, though it costs a bit more messages and longer time.

Keywords: Mutual Exclusion, Distributed Algorithm, MANET, Mobile Computing, Token Loss.

1 Introduction

Mutual exclusion (MUTEX) is a fundamental problem in distributed systems, where a group of hosts intermittently require entering the Critical Section (CS) in order to exclusively perform some critical operations, e.g. accessing the shared resource. A solution to the MUTEX problem must satisfy the following three correctness properties:

- *Mutual Exclusion* (safety): At most one host can be in the CS at any time;
- *Deadlock Free* (liveness): If any host is waiting for the CS, then in a finite time some host enters the CS;
- *Starvation Free* (Fairness): If a host is waiting for the CS, then in a finite time the host enters the CS.

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 572–583, 2007.

Many MUTEX algorithms have been proposed for traditional distributed systems [19], which can be categorized into two classes: *token-based* algorithms and *permission-based* algorithms. In token-based algorithms, a token is passed among all the hosts. A host is allowed to enter the CS only if it possesses the token. In a permission-based algorithm, the host requesting for the CS must first obtain the permissions from other hosts by exchanging messages.

With the emergence of mobile networks, new challenges are introduced in solving the MUTEX problem [6][17]. In this paper, we focus on mobile ad hoc networks (MANETs), which have become the focus of research in recent years. MANETs have fundamentally different properties from traditional wired networks in the aspects of communication, mobility and resource constraint. In a MANET, each mobile host (MH) plays the same role and the communication channels between hosts are multiple hops in nature. The topology of a MANET is arbitrary and can change dynamically. These characteristics should be seriously considered in the design of a MUTEX algorithm.

During the past years, several MUTEX algorithms for MANETs have been proposed and nearly all of them use the token-based approach [1][2][9][20]. Compared with the permission-based approach [22], the token-based approach has many desirable features, e.g. hosts only need to keep the information about their neighbors and few messages are needed to pass the privilege of entering CS. Both token-circulating (ring-based) and token-asking (tree-based or graph-based) approaches have been used in MUTEX algorithms for MANETs. Compared with the token-asking approach, where a host needs to send its request for CS to the token holder, token-circulating has two advantages. First, a logical ring is much simpler to maintain than a tree or graph. Second, besides the token itself, no other control message, e.g. the request message, is needed in a token-circulating algorithm. Therefore, the token-circulating approach is considered more suitable for MANETs.

However, token-based algorithms may suffer from the token loss problem, which becomes more serious in MANETs due to mobility, weak communication links and weak hosts. Unfortunately, existing token-based MUTEX algorithms for MANETs focus on investigating the mechanism of establishing the logical structure, but have not adequately addressed the token loss problem. In traditional wired networks, timeout-based mechanisms are commonly used to detect token losses [4][8][10][12][16][18]. However, in MANETs, the mobility and frequent disconnections of MHs increase both the probability of token loss and the difficulty in setting a proper timeout value.

In this paper, we propose the MDTM (Mobile Dual-Token MUTEX) algorithm, a timeout-free MUTEX algorithm for MANETs, which tolerates token losses using a dual-token approach. The dual-token approach has been used in traditional networks. For example, J. Misra has proposed a token loss detection mechanism, hereafter called JM [11], where two tokens monitor each other to detect token losses without using timeout. However, JM has two problems when it is applied to MANETs. First, it requires maintaining the FIFO (Fist-In-First-Out) channel between non-neighboring MHs, which can be costly and even unfeasible in a MANET due to the dynamics of the network. Second, using a static ring to circulate the tokens is inefficient. In the design of the MDTM algorithm in this paper, we propose techniques to solve the above two problems and develop the dual-token approach that is suitable to MANETs.

The rest of the paper is organized as follows. Section 2 reviews existing MUTEX algorithms for MANETs and briefly describes the background knowledge on the dual-token approach. The detailed description of our proposed MDTM algorithm, including the system model, data structures and operations, is presented in Section 3. We present the correctness proof and performance evaluation of MDTM in Section 4 and Section 5 respectively. Finally, Section 6 concludes the paper with discussions on our future work.

2 Related Work and Background

Several MUTEX algorithms have been proposed for MANETs, nearly all of which use token-based approaches. To our knowledge, the only permission-based MUTEX algorithm for MANETs is our previous work reported in [22], which enforces MUTEX among only the hosts currently competing for the CS so as to reduce the message cost. In this section, we focus on token-based algorithms, including token-asking algorithms and token-circulating algorithms.

Token-asking algorithms are proposed in [20][21], which adopt the tree-based approach proposed in [15]. A directed acyclic graph of token-oriented pointers is used to maintain multiple paths leading one MH to the token holder. Like in [15], the token (the privilege of entering CS) and requests for executing the CS are passed along the paths in the graph. Token losses are handled by the detection of link breaks, but how to detect link breaks is not discussed. Token-circulating algorithms for MANETs are proposed in [1][5]. To save communication cost, the ring, along which the token is passed, is computed on-the-fly.

As discussed before, token-based algorithms, especially token-circulating algorithms, have some desirable features for MANETs, but token loss is a big problem. Usually, timeout-based mechanisms are used to detect token losses [4][8][10][12][16][18]. In MANETs, However, the mobility and frequent disconnections of MHs increase both the probability of token loss and the difficulty in setting a proper timeout. In this paper, we propose a dual-token approach to detect token losses in MANETs, which does not need to use timeout.

Previous work also investigated the dual-token approach for traditional wired networks, e.g. the JM algorithm [11]. There are two symmetrical tokens in the system, while only one of them carries the privilege of executing CS. By comparing the sequence number of the token and the one kept by a host, a token can determine whether the other token is lost. The JM algorithm is simple and timeout-free, which are desirable features for MANETs. However, JM has two problems when it is applied in a MANET.

First, JM requires the channel between any pair of hosts to be a FIFO channel. A non-FIFO channel may cause a false detection of token loss. Since the successor/predecessor of a host in a logical ring may not be a physical neighbor, FIFO channels between non-neighboring hosts are necessary in JM. In a MANET, however, because of movements and disconnections of MHs, the network topology changes frequently, and, therefore, it is difficult to maintain a FIFO channel between two MHs that are not physical neighbors.

Second, JM uses a static ring to circulate the two tokens, which is not efficient in MANETs. To reduce the communication cost, one commonly used approach to cope with this problem is computing the ring on-the-fly, i.e. determining the successor of a host dynamically [1][5][9]. However, if the ring in JM is constructed on-the-fly, the two tokens may be passed along different rings, which may result in a false detection of token loss. For example, there are three hosts, m_i, m_j and m_k. The last token that visited m_k is *T'* and the two tokens do not encounter each other after *T'* visited m_k last time. After visiting other hosts, m_i directly sends *T* to m_k but it sends *T'* to m_k through m_j due to the dynamically selected successor. Then, *T'* may arrive at m_k before *T* regardless the sequence of sending tokens. Consequently, m_k falsely claims the loss of *T*, which may be actually held by m_j.

3 The MDTM Algorithm

Using a dual-token approach, we design a new distributed MUTEX algorithm for MANETs without requiring the use of timeout.

First, to relax the requirement on FIFO channels between non-neighboring hosts, we use a different method to determine the token loss. Each token carries a flag which indicates the state of another token and a coordinator is assigned in each round of the circulation of a token. On the reception of the token, a host checks the sequence numbers of the token and its own. If a token holder is not the coordinator of the current round, it only updates the token flag. The determination of a token loss and the regeneration of a new token are entailed on the coordinator. When one round of circulation finishes, the token is sent back to the coordinator of the round, which determines whether the other token has been lost and, if so, regenerates a new one.

Second, our method of token loss detection can also help relax the dependence on the static ring. In MDTM, the two tokens are circulated separately along their own rings. The successor on the ring is computed on-the-fly.

In the rest of this section, we describe the detailed design of the MDTM algorithm, including the system model, data structures and operations.

3.1 System Model

We consider a MANET that consists of a collection of *n* autonomous MHs, $M = \{m_1, m_2, \ldots, m_n\}$. The hosts may fail by crashing but they recover after a finite time. Each MH has at most one request for the CS at any given time.

The MHs communicate with each other by exchanging messages through wireless channels. Each host is a router and the communication between two hosts can be multiple hops. Whether two hosts are directly connected is determined by the signal coverage range and the distance between the hosts. If two MHs are directly connected, they are called neighbors and the communication channel between them is called a link. A link may fail for various reasons. Due to the node movements and failures, the topology of the MANET can change arbitrarily.

To guarantee the correctness of MDTM, the link between neighboring MHs is assumed to satisfy the FIFO property. In wireless communication, all the links between a host and its neighbors share the same communication medium, i.e. a host communicates with all its neighbors using in fact the same channel. Therefore, we assume that all the messages transmitted between a host and all its neighbors satisfy the FIFO property.

3.2 Data Structures and Message Types

The data structures used in MDTM are as follows:

T_p, T_s: the primary and secondary token. Each of them has four fields: *c*, *ft*, *co*, and *D*. *c*: the sequence number of the token. It is a positive (negative) value for *Tp* (*Ts*). *ft*: a flag used to indicate whether the other token is alive. *co*: the coordinator of the current round. It can be initialized to any host. *D*: the set of hosts to be visited by the token in the current round.

Initially, T_p and T_s are initialized to {1, *false*, *n*-1, *M*} and {-1, *false*, *n*-1, *M*} respectively. Obviously, the *Tp.co* and *Ts.co* can be initialized to any value between 0 and *n*-1. Here, for simplicity, we choose *n*-1.

ct_i: a sequence number of a host m_i, which is equal to the sequence number of the last token received. Initially, it is set to 0.

Only one type of message is used in the MDTM algorithm:

TOKEN (T, tt): the message to transfer a token. *T* is the token transferred and *tt* is the type of the token, whose value can be 1(for T_p) or 2(for T_s).

3.3 Operations of MDTM Algorithm

The two tokens T_p and T_s are circulated separately. The circulation of a token is divided into rounds. During each round a MH is declared to be the coordinator and the token visits each MH one time. To balance the workload of MHs, the coordinator of a new round is dynamically selected by choosing the host nearest[1] to the current coordinator.

To reduce the communication cost, the ring, along which a token is circulated, is dynamically constructed according to the underlying network topology. A host does not determine its successor host until it is holding a token and wants to send it out. The token needs to keep the IDs of the hosts which have not been visited in the current round. Fig. 1 shows an example ring.

The pseudocode of the MDTM algorithm is shown in Fig. 2. As mentioned earlier, the two tokens are circulated separately and asynchronously. A new round of a token *T* starts from the coordinator of the current round[2] m_c. All MHs are put in the target set *T.D*, and the nearest MH to m_c is selected as the new coordinator. The change of the coordinator is for balancing the load of the hosts. Then, m_c sends the token to the new coordinator using the message *TOKEN*(*T, tt*). It is important to notice that each token has its own ring and coordinator.

Upon the reception of a *TOKEN (T, tt)* message, the receiver m_i first checks whether it is also holding the other token. If so, $T_p.c$ and $T_s.c$ are respectively increased or decreased by one to prevent the false detection of a token loss after the tokens met at the same host. Otherwise, m_i compares *T.c* and ct_i. If $ct_i \neq T.c$, which

[1] Here, the distance is examined using the number of hops between two MHs.

[2] The first round starts from an arbitrarily selected host.

means that the other token ever visited m_i after T visited m_i last time, m_i sets the flag *T.ft* to *true*. Then, m_i sets its sequence number ct_i to *T.c* and removes its ID from *T.D*. If T is the primary token T_p and m_i has a pending request, m_i executes CS.

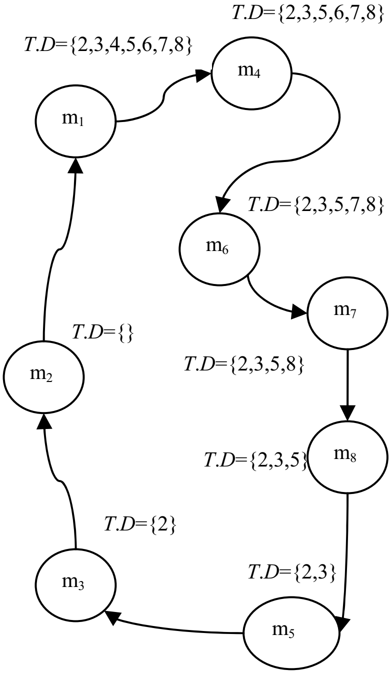


Fig. 1. An Example Ring

```
COBEGIN:// The code executed by each host, $m_i$
  Upon the reception of a message $TOKEN(T, tt)${
    if(both tokens held by $m_i$){
      $T_p.c \leftarrow T_p.c+1$;
      $T_s.c \leftarrow T_s.c-1$;
    }
    if($ct_i \neq T.c$)
      $T.ft \leftarrow true$;
    $ct_i \leftarrow T.c$;
    remove $m_i$ from $T.D$;
    if ($tt$ = 1and $m_i$ is requesting CS)
      execute CS;
    if($i$=$T.co$ and $T.D$= $\varphi$){
      if($T.ft$=$false$){
        $T.c \leftarrow T.c+1$;
        regenerate the other token $T'$;
        $T'.c \leftarrow -T.c$; $T'.co \leftarrow$the nearest in $T'.D$;
        send $TOKEN(T', tt)$ to nearest in $T'.D$;
      }else{
        $T.D \leftarrow M$;
        $T.co$ $\leftarrow$the nearest in $T.D$;
        $T.ft$=$false$;
      }
    }
    if($T.D \neq \varphi$)
      send $TOKEN(T, tt)$ to the nearest in $T.D$;
          else send $TOKEN(T, tt)$ to $T.co$;
  }
COEND
```

Fig. 2. The Operations of the MDTM Algorithm

Following the possible execution of CS, m_i sends T to the nearest host in *T.D* using a message *TOKEN* (*T, tt*) if m_i is not the coordinator of T in current round or T has not finished the circulation of the current round (*T.D* ≠). Otherwise, if m_i is the coordinator of T in the current round and T has finished the circulation of the current round (*T.D* =), m_i performs the following operations depending on the flag *T.ft*.

Case 1: *T.ft* = *false*. This indicates that the other token *T'* did not visit any host after the last visit of T. Obviously, *T'* is lost. m_i increases (for T_p) or decreases (for T_s) *T.c* by one and regenerates *T'*. *T.D* and *T'.D* are set to *M*. A new round starts for both tokens.

Case 2: *T.ft* ≠ *false*. This means that the other token *T'* ever visited some host after the last visit of *T*. *T.D* is set to *M* and a new round for *T* starts.

Now, let us consider the crash of a host. When a host crashes, it stops doing any operation, e.g. executing the CS or sending a message, until it recovers after some finite time. Therefore, a crash may cause the execution of the MDTM algorithm blocked for some finite time. On the recovery from the crash, the host simply resumes its execution without doing any special operation. If the host has a stable storage to store its execution state, it can resume the execution from its last state before the crash; otherwise, it can resume from the initial state.

4 Correctness of the MDTM Algorithm

In this section, we prove the correctness of the proposed algorithm by showing that the three correctness requirements for distributed MUTEX algorithms are satisfied.

Safety. Since only a MH holding the token T_p can enter the CS, the safety property can only be violated when there is a false detection of the loss of T_p, i.e. a living T_p is perceived to be lost. Therefore, we only need to show that if T_p is perceived to be lost by MDTM, T_p must be really lost.

In MDTM, the loss of T_p is indicated by the flag $T_s.ft$ of the secondary token T_s. From the operations of MDTM we know that, T_p is perceived to be lost only if the sequence numbers of all MHs are the same as the sequence number of T_s. By the assumption of the FIFO channel between physical neighboring MHs, such a scenario can happen only if: 1) T_p does not visit any host between two consecutive rounds of T_s, and 2) T_p is not in any channel. This is a sufficient condition for the loss of T_p. Therefore, MDTM guarantees the safety property.

Liveness. If the primary token T_p is not lost, the liveness property of MDTM is obviously guaranteed. If T_p is lost, it is regenerated once the loss is detected. Therefore, we only need to show that if T_p is lost, the loss must be detected by MDTM. In the following proof, we assume that the system will not lose the two tokens simultaneously, i.e. at least one token is alive at any moment[3].

If T_p is lost in some round, it can no longer visit any host. Then, the sequence number of T_s is no longer changed. After T_s finishes the round that starts after the loss of T_p, the sequence numbers of all MHs are set to the same as the sequence number of T_s. Then, T_s starts a new round and $T_s.ft$ is set to *false*. During this round, no MH changes the value of $T_s.ft$. Therefore, when T_s reaches the coordinator at the end of this round, T_s detects the loss of T_p and regenerates T_p. Therefore, MDTM guarantees the liveness property.

Fairness. If T_p is not lost, T_p visits each MH at least once in each round. Therefore, when a MH generates a request for CS, it can enter CS in the current or next round of T_p. If T_p is lost, by the liveness property of MDTM, T_p will be regenerated after T_s detects the loss of T_p. Therefore, if a MH has a pending request for CS, it can enter CS after some finite time. The fairness property is guaranteed.

5 Performance Evaluation

We have conducted simulations to evaluate the performance of our proposed algorithm. To compare with timeout-based approach, we also simulated a simplified version of the algorithm proposed in [1] by removing the token-asking mechanism.

[3] If both the tokens can be lost at the same time, more than two tokens are necessary to guarantee the liveness property. Please see Section 6 for more discussions.

We denote this algorithm by DYNT (DYNamic Token-circulating). To detect token losses, we let a host in DYNT set a timeout for the token. Once a timeout happens, the token is perceived to be lost and a new toke is generated. Same as the MDTM algorithm, the ring in DYNT is computed on-the-fly to save communication cost.

5.1 Simulation Setup and Metrics

The simulation system consists of two modules: the network and the MUTEX algorithm. The main parameters of the simulations are shown in Table 1. There are totally 20 MHs in the system and they are randomly scattered in a rectangular territory. The MHs move according to the well-known random waypoint mobility model [3]. The mobility level, defined as the percentage of the time that a host does move over the total time, is fixed to 50%.

For message routing, we implemented a simple protocol based on the "least hops" policy, which is adopted in many classical routing protocols in MANETs [7][13][14]. A routing table is proactively maintained at each host. The message delay is assumed to satisfy the exponential distribution. However, to guarantee the FIFO property between a MH m_i and its physical neighbors, a message arriving at m_i may not be delivered until it becomes the earliest one among all the messages received on the same channel but not delivered to m_i. The message loss rate is varied from 0% to 20%.

Table 1. Parameters of Simulations

Number of Hosts	20
Territory scale	224m
Mobility model	Random-waypoint
Mobility level	50%
Transmission radius	100m
Routing protocol	Least hops
Message loss rate	0% to 20%
Execution time of CS	0~50ms
Load level,	1.0E+0, 1.0E-1, 1.0E-2
Simulation time	10 hours

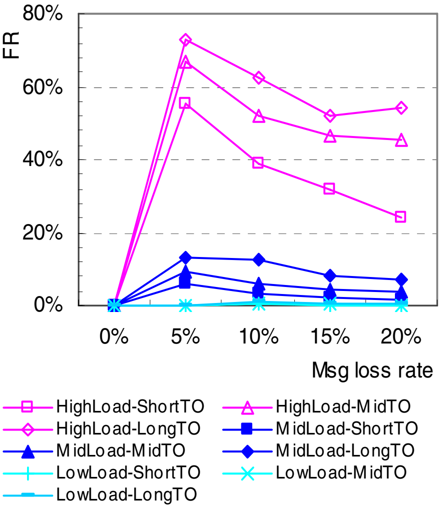


Fig. 3. FR vs. message lost rate

The MUTEX algorithms are implemented as applications running at MHs. The arrival of the requests at a host is assumed to follow the Poisson distribution with mean , which represents the number of requests generated by a single host per second. We simulated three different load levels, high level (=1.00E+0), middle level (=1.00E-1) and low level (=1.00E-2). The duration of the execution of CS is assumed to follow the uniform distribution from 0ms to 50ms. The timeout for the token is the key parameter for DYNT. If no message is lost, no timeout needs to be set. In the cases with message losses, we set the timeout value according to the response time (see the definition below) of the corresponding cases without message loss. To examine the effect of different timeout values, we adopt three timeout levels: long timeout (twice of the corresponding response time), middle timeout (five times of the corresponding response time), and short timeout (ten times of the corresponding response time).

In the simulations, we measure the performance of the algorithms using the following metrics:

False token loss detection Rate (*FR*): The number of token losses falsely claimed over the total number of token losses detected. This metric is used to examine the accuracy of token loss detection mechanisms.

Number of Hops per CS entry (*NH*): The average number of hops of all the messages exchanged for each entrance of the CS. One hop means a network layer message, i.e. the point-to-point message.

Response Time (*RT*): The time that a host waits to enter the CS after its request for CS is generated.

5.2 Simulation Results

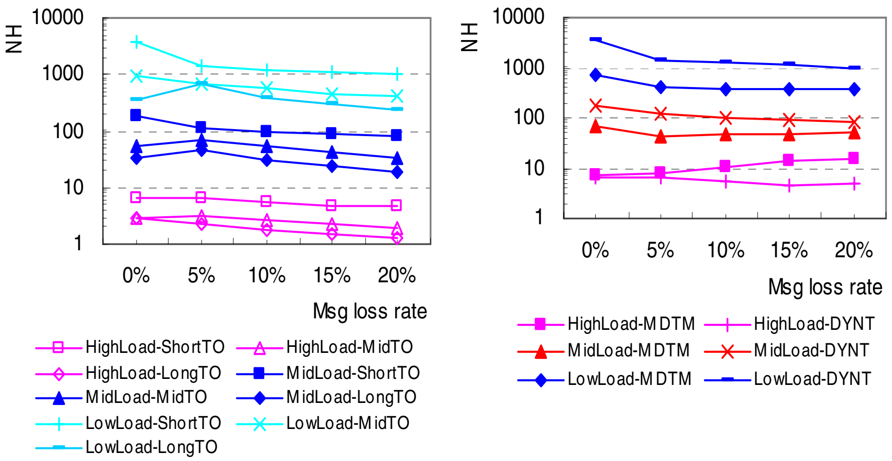
1) *False token loss detection Rate, FR*

The simulation results are presented as follows. Fig. 3 shows FR under various conditions. For our MDTM algorithm, FR is always equal to zero, which indicates that MDTM can precisely detect token losses. Therefore, only the FR of DYNT is depicted in Fig. 3.

For the DYNT algorithm, if no message is lost, there is no false detection because no timeout needs to be set and therefore no token loss is detected at all. If message losses occur, DYNT may falsely detect token losses in nearly all the cases. The higher the load level is, the higher the FR is. Under high load levels, each MH always has a pending request, so each MH will execute the CS when it obtains the token T_p. Therefore, the RT under high load levels is much larger and varies stronger than that under low load levels (as shown in Fig. 5). Consequently, more timeouts occur and FR is increased.

Another factor affecting FR is the timeout value. Generally, a longer timeout causes a larger FR, especially when the load level is high. This can be explained as follows. When the timeout is large, a long time is needed to detect a token loss and, consequently, the response time is prolonged (as shown in Fig. 5). Therefore, more timeouts occur, which results in a larger FR. In the simulated cases, the short timeout is better than the other two. Of course, this does not mean that a small timeout is always better than a large one. A too small timeout will certainly cause more false detections. As discussed before, determining a proper timeout value is a challenging issue. This is why we adopt a timeout-free approach in the design of MDTM.

The effect of message loss rate is more complex. In general, with the message loss rate changes from 5% to 20%, FR of DYNT decreases slowly. The effect of message loss rate is twofold. Intuitively, the more messages are lost, the more timeouts may be triggered and, consequently, the more false detections may occur. However, a high message loss rate results in a large RT, as discussed below. Since a MH only generates a new request after its previous request has been met, a large RT implicitly decreases the actual request arrival rate, i.e. the load level, at a MH. With the effect of load level as discussed above, FR of DYNT decreases with the increase of the message loss rate.



(a) (b)

Fig. 4. NH vs. message loss rate

2) *Number of Hops per CS entry, NH*

The message cost of DYNT is shown in Fig. 4-(a). As in most token-circulating algorithms, the lower the load level is, the higher probability the transmission of the token does not serve any CS entrance. The timeout value also affects the NH of DYNT. NH under a long timeout is smaller than that with a short timeout. The underlying reason is that, FR under a longer timeout is also larger as shown in Fig. 3, so more redundant tokens are generated, which can significantly reduce NH. Fig. 4-(a) shows that the message loss rate does not affect NH obviously.

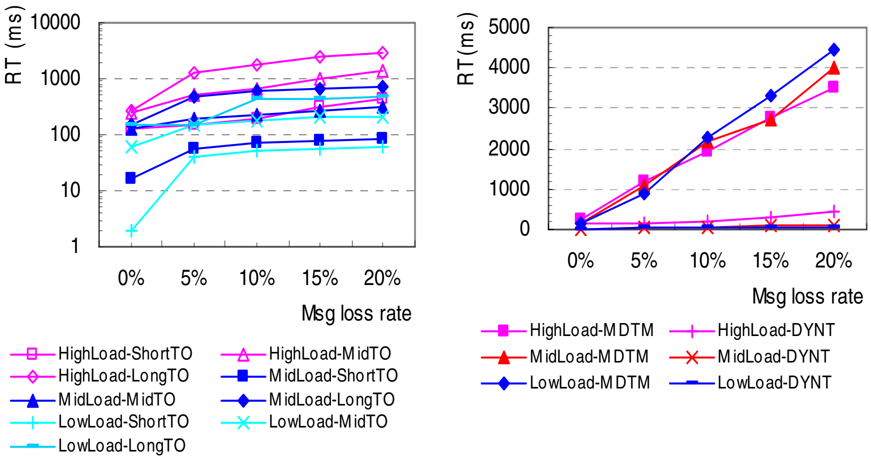
Fig. 4-(b) shows the NH of our MDTM algorithm in comparison with DYNT with a short timeout. In general, our MDTM algorithm costs more hops for each CS entrance. This is due to the dual-token approach. Circulating two tokens needs more messages than one token.

3) *Response Time, RT*

Fig. 5-(a) shows the RT of DYNT under various conditions. Fig. 5-(b) shows the RT of MDTM in comparison with DYNT with a short timeout. In general, with the increase of the load level, the message loss rate, or the timeout value, RT also increases. This is easy to understand.

Now, let us compare DYNT and MDTM. If there is no message loss, the two algorithms perform nearly the same. However, if message losses occur, a MH needs to wait for a longer time in MDTM than in DYNT. This should be caused by the false token loss detections of DYNT as shown in Fig. 3. Once a false detection occurs, the redundant token T_p may appear, which will significantly reduce RT. Obviously, a false detection will cause the violation of the mutual exclusion property of a MUTEX algorithm.

In summary, although our dual-token mechanism may have a high time/message cost, it can precisely detect token losses and, therefore, guarantee the mutual exclusion property, which is essential and must be achieved.



(a) (b)

Fig. 5. RT vs. message loss rate

6 Conclusions and Discussions

In this paper, we proposed a dual-token based MUTEX algorithm to address the token loss problem in MANETs. There are two symmetrical tokens, which monitor each other using sequence numbers to detect token losses. Compared with timeout-based algorithms, our proposed algorithm is more suitable for MANETs where it is difficult to set a proper timeout for a token due to host movements and disconnections. To reduce the communication cost of passing the tokens, the ring is computed on-the-fly. Simulation results show that our proposed algorithm can precisely detect the token loss and guarantee the mutual exclusion property although the time and message costs can be high.

One problem of the MDTM algorithm is that, if both the tokens are simultaneously lost, the liveness property cannot be guaranteed. To address this problem, more tokens need be used. These tokens can be ordered in a ring, and each token monitors its successor. Another improvement that can be made is combining the token-asking mechanism to improve the performance under low load levels. Of course, some synchronization is necessary to synchronize the operations of the two tokens.

Acknowledgments. This research is partially supported by Hong Kong University Research Grant Council under the CERG grant PolyU 5183/04E, France/Hong Kong Joint Research Scheme F-HK16/05T, and the China National "973" Program Grant 2007CB307100

References

1. Baldoni, R., Virgillito, A., Petrassi, R.: A Distributed Mutual Exclusion Algorithm for Mobile Ad-Hoc Networks. In: Proc. of ISCC (2002)
2. Benchaïba, M., Bouabdallah, A., Badache, N., Ahmed-Nacer, M.: Distributed Mutual Exclusion Algorithms in Mobile Ad Hoc Networks: an Overview. ACM SIGOPS Operating Systems Review 38(1) (2004)
3. Camp, T., Boleng, J., Davies, V.: A Survey of Mobility Models for Ad Hoc Network Research. Wireless Communications & Mobile Computing (WCMC) 2(5) (2002)
4. Chang, Y., Singhal, M., Liu, M.: A Fault Tolerant Algorithm for Distributed Mutual Exclusion. In: Proc. of SRDS (1990)
5. Chen, Y., Welch, J.: Self-stabilizing Mutual Exclusion Using Tokens in Ad Hoc Networks. In: Proc. of Dial-M (April 2002)
6. Forman, G., Zahorjan, J.: The Challenges of Mobile Computing, IEEE Computer (1994)
7. Johnson, D., Maltz, D.: Dynamic Source Routing in Ad Hoc Wireless Networks, Mobile Computing. Ch. 5. Kluwer Academic Publishers, Boston, MA (1996)
8. Le Lann, G.: Distributed Systems, towards a Formal Approach, IFIP Congress (1977)
9. Malpani, N., Vaidya, N., Welch, J.: Distributed Token Circulation on Mobile Ad Hoc Networks. In: Proc. of ICNP (2001)
10. Mizuno, M., Neilsen, M., Rao, R.: A Token based Distributed Mutual Exclusion Algorithm based on Quorum Agreements. In: Proc. of ICDCS (1991)
11. Misra, J.: Detecting Termination of Distributed Computations Using Markers. In: Proc. of PODC (1983)
12. Nishio, S., Li, F., Manning, G.: A Resilient Mutual Exclusion Algorithm for Computer Networks. IEEE Trans. on Parallel and Distributed Systems 1(3) (July 1990)
13. Perkins, C., Bhangwat, P.: Highly Dynamic Destination-Sequenced Distance-Vector (DSDV) Routing for Mobile Computers. In: Proc. of ACM SIGCOMM 1994 (August 1994)
14. Perkins, C., Royer, E.: Ad-hoc On-Demand Distance Vector Routing. In: Proc. of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (February 1999)
15. Raymond, K.: A Tree-based Algorithm for Distributed Mutual Exclusion. ACM Transactions on Computer Systems 7(1) (1989)
16. Sanders, B.: The Information Structure of Distributed Mutual Exclusion Algorithms. ACM Trans. on Computer Systems 5(3) (1987)
17. Satyanarayanan, M.: Fundamental Challenges in Mobile Computing. In: Proc. of PODC (1996)
18. Singhal, M.: A Heuristically-aided Algorithm for Mutual Exclusion in Distributed Systems. IEEE Trans. on Computers 38(5) (1989)
19. Singhal, M. : A Taxonomy of Distributed Mutual Exclusion, Journal of Parallel and Distributed Computing (18) (1993)
20. Walter, J., Kini, S.: Mutual Exclusion on Multihop, Mobile Wireless Networks, Texas A&M Univ., College Station, TX 77843-3112, TR97-014 (December 9, 1997)
21. Walter, J., Welch, J., Vaidya, N.: A Mutual Exclusion Algorithm for Ad Hoc Mobile Networks. Wireless Networks 9(6) (November 2001)
22. Wu, W., Cao, J., Yang, J.: A Scalable Mutual Exclusion Algorithm for Mobile Ad Hoc Networks. In: Proc. of ICCCN (2005)

Study of a Cost-Effective Localization Algorithm in Wireless Sensor Networks[*]

Xin Li, Bei Hua[**], and Yan Guo

Department of Computer Science and Technology
University of Science and Technology of China, Hefei, Anhui 230027, China
Mobile Computing Laboratory
Suzhou Institute for Advanced Study, Suzhou, Jiangsu 215123, China
xinxinol@mail.ustc.edu.cn,bhua@ustc.edu.cn,guoyan6@mail.ustc.edu.cn

Abstract. Most of the current RSS (Received Signal Strength)-based localization algorithms in Wireless Sensor Networks (WSNs) rely on isotropic radio propagation model to infer the distance between a pair of nodes from received signal strength, which however has been proved to be quite unreliable in recent research work. Performance analysis is important to evaluate the applicability of a localization algorithm, however little work has been done on evaluating the existing localization algorithms in simulated realistic settings. This paper firstly presents the motivation and detailed implementation of a proposed Link-State Based Annulus (LSBA) localization algorithm, and then gives a panorama of performance comparison among LSBA and other four localization algorithms in terms of estimation error, convergence speed, computational complexity and communication cost in the simulated realistic environment. Simulation results show that LSBA achieves the best tradeoff among all the four metrics in WSNs with moderate number of anchors, and has good adaptability to irregular node deployment as well.

Keywords: RSS; wireless sensor networks; localization.

1 Introduction

Localization approaches in WSNs are roughly classified as fine-grained approaches and coarse-grained approaches. Fine-grained approaches normally require accurate distance or angle measurements to compute the location of unknown node. TDOA (Time Difference of Arrival) [7] [8] [9], AOA (Angle of Arrival) [10], and RIPS (Radio Interferometric Positioning System) [12] rely on extra (sometimes expensive and complex) hardware other than radio transceiver to get accurate measurements. Use of RSS as ranging technique receives much recognition since radio transceiver is the only available ranging device for most of the common sensor nodes. Most of the existing RSS methods rely on an ideal

[*] This work was supported by the National Natural Science Foundation of China under Grant No.60673173.

[**] Corresponding author.

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 584–595, 2007.

radio propagation model [16] to get the distance from RSSI, which states that the received signal strength diminishes with the distance according to certain law. However, recent researches [13] [14] [15] show that radio propagation pattern is highly random in real world, and no one-to-one mapping exists between RSSI and distance in most situations. Therefore localization algorithms based on ideal radio model may perform poorly in realistic environment, and need to be reconsidered. Coarse-grained approaches normally rely on proximity and near-far information or less accurate distance estimation to infer the location of unknown node, of which Centroid [2], geometry constrains [6], DV-HOP [4], Amorphous [5] are typical. We provide a LSBA (Link State Based Annulus localization algorithm) [17]. Coarse-grained approaches are much popular in densely deployed large scale sensor networks since they avoid the difficulty of getting accurate measurements; however their performance may degrade due to inaccurate information and need careful evaluation.

Performance of a localization algorithm can be measured by various metrics, in which estimation error, un-localizable ratio, convergence speed, computational and communication complexity are the most important ones. Average estimation error is a classical performance metric that measures the average localization accuracy. Un-localizable ratio is the ratio of nodes that cannot be localized even after certain rounds of localization, which can be used to depict how fast the process winds up with stable estimation error. Since wireless sensor nodes are tiny devices with constrained computing ability, storage, bandwidth, and energy, computational complexity must be taken into account. The requirement of energy saving is usually translated into lowering the communication complexity, since most of the energy is consumed by communication in WSN. However, most of the previous work only take one or two of the performance metrics into consideration, e.g., [2] [4] [5] and [6] only focus on estimation accuracy; [3] and [17] focus on estimation accuracy, communication cost or un-localizable ratio. In this paper, we provide a comprehensive performance comparison of five algorithms in terms of the four performance metrics in realistic experimental settings.

The remainder of this paper is organized as follows: section 2 outlines the motivation of LSBA; section 3 describes the detailed implementation of LSBA; section 4 compares LSBA with Centroid [2], APIT [3], DV-HOP [4], and Amorphous [5] in various performance metrics in realistic experimental settings; and section 5 concludes.

2 Motivation

Our work is largely inspired by [14], where a Radio Irregularity Model (RIM) is brought up based on empirical data obtained from MICA2 platform to characterize the radio pattern in real wireless sensor networks. Radio irregularity is mainly caused by heterogeneous sending powers and anisotropic path losses, and is modeled as follows:

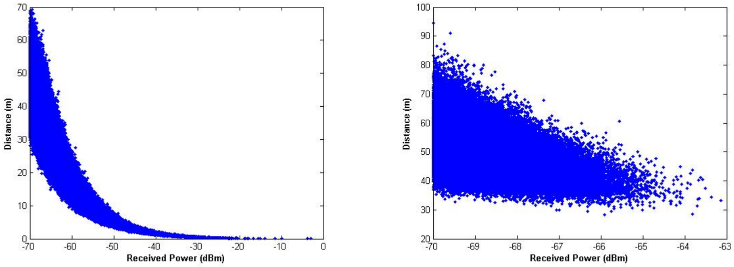
$$RSS = SP \times (1 + R \times VSP) - PL \times K_i + X \tag{1}$$

The first part of (1) accounts for the difference in hardware calibration and battery status, where SP (Sending Power) is the power rating of node, VSP (Variance of Sending Power) is defined as the maximum percentage variance of signal sending power among different devices, and R is a normal random variable that measures the variance caused by hardware. The second part reflects the anisotropism of radio, where PL is the free space loss, and K_i represents the difference in path loss in different directions.

$$K_i = \begin{cases} 1, i=0 \\ K_{i-1} \pm Rand \times DOI, 0 < i < 360 \wedge i \in N \end{cases} \tag{2}$$
$$where \ \ |K_0 - K_{359}| \leq DOI$$

K_i is calculated according to (2), where DOI (Degree of Irregularity) is defined as the maximum range variation per unit degree, and Rand is a Weibull distribution random variable. The third part of (1) models the environment noise that follows normal distribution. More details can be found in [14].

Experiments show that lots of asymmetric links exist in network due to radio irregularity, and the number of asymmetric (symmetric) links increases (decreases) with longer distance. This observation inspired us to exploit link information to reduce the distance uncertainty between a pair of nodes. To validate our imagination, we simulated with RIM model in various parameter settings and got two curves on RSS vs. distance for symmetry links and asymmetry links respectively. Fig.1 was obtained with SP=-5dBm, VSP=0.1, DOI=0.002, Weibull=[0.16, 0.67], and receiver threshold=-70dBm.



(a) Symmetry link

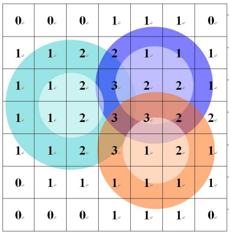
(b) Asymmetry link

Fig. 1. RSS vs. Distance

The curves suggest that rough relationships exist between RSSI and range of distance when link type is given. This result motivated us to design LSBA that makes use of the link type and RSSI to refine the possible areas an unknown node may reside in, and then calculates the centroid of the overlapping area as the location. Since the curves of RSS vs. Distance vary with network settings and environment, they need to be recomputed for each concrete situation. Nevertheless, it will not add much difficulty, since all the device related parameters such as SP, VSP and receiver threshold can be easily obtained from devices, and the environmental parameters such as DOI and Weibull distribution can be obtained from empirical values or measured in real environment.

3 Implementation of LSBA

The basic idea of LSBA is as follows: each unknown node exchanges messages with its location-aware neighbors to find out the type of link (symmetric or asymmetric) between itself and each of these neighbors; then calculates the range of distance to each neighbor based on the type and RSSI of each link; after getting all the distance ranges, draws an annulus for each link centering at the location-aware node with the corresponding distance range; finally calculates the centroid of the highest overlapping area as its location estimation. Therefore LSBA consists of two steps: link state construction and location calculation, in which a link state exchange protocol and a grid-based method are provided to simplify the calculation of the highest overlapping area of annuli (fig.2(a)).



0	0	0	1	1	1	0
1	1	2	2	1	1	1
1	1	2	3	2	2	1
1	1	2	3	3	2	2
1	1	2	3	1	2	1
0	1	1	1	1	1	1
0	0	0	1	1	1	0

(a) Annuli Drawing

RSSI (dBm)	-5	...	-60	...	-70
Min (m)	0.0000	...	10.0129	...	28.9374
Max (m)	0.0000	...	30.3575	...	70.1853

(b) RSS vs. Distance lookup table of fig.1(a)

Fig. 2. Implementation of LSBA

The link state exchange protocol works as follows. Each node broadcasts its ID and location (if it's a location-aware node) in an advertisement message. After receiving an advertisement, a node records the sender and RSSI of the message and puts [sender, RSSI] in its Asymmetry-RSSI-Array, then calculates the distance range (expressed as a [min, max] pair) to the sender using the curve for asymmetric link and puts [sender, [min, max]] in the Asymmetric-Distance-Array, at last sends back a response message containing the sender and the intended receiver. If an asymmetric link exists between a pair of nodes, say from A to B, then only B receives A's advertisement and A cannot hear B, and the message exchange between A and B ends at this point. If a symmetric link exists between a pair of nodes, then either side can receive the advertisement and response message of the other side. After receiving a response message that replies to its previous advertisement, a node removes the corresponding item of [sender, RSSI] from the Asymmetry-RSSI-Array to the Symmetry-RSSI-Array, and deletes the corresponding item of [sender, [min, max]] from the Asymmetric-Distance-Array, then recalculates the distance range using the curve for symmetric link and puts the new [sender, [min, max]] in the Symmetric-Distance-Array, at last sends back a distance notification message containing the distance range it calculates. The exchange of distance range is to further reduce the uncertainty of distance between a pair of nodes. Therefore after receiving a distance notification, the unknown node takes the intersection of the two distance ranges as the final distance range. The current link state exchange protocol doesn't take into account collision and packet lost.

To simplify the computation of distance range, we use two lookup tables calculated offline to replace the two RSS vs. Distance curves. Each table is organized as an array, where N is the size of the table and each entry records the [min, max] pair of a RSSI. As an example, fig.2(b) approximates the curve in fig.1(a) with the range of RSSI from -5dBm to 70dBm and a step size of 1dBm. For a non-integral RSSI, the nearest integral RSSI is taken.

If there are only a few anchors in the network, many nodes may remain unknown after the first round of localization. In the following rounds, all location-aware nodes, including anchors and those getting their locations in the previous round, broadcast their locations, and non-anchor nodes either locate themselves or refine the locations obtained in the previous rounds. This process repeats until the algorithm converges. The decision of when to stop is made in a distributed way. When a non-anchor node finds that the variance of its location estimation is below a threshold, it stops updating its location and stops broadcasting advertisement, but it can still respond to other node's advertisement. When all the nodes cease broadcasting advertisement, the localization process winds up.

4 Performance Evaluation

We simulated Centroid, APIT, DV-HOP, Amorphous and LSBA on Matlab and compared them in terms of estimation error, residual ratio of un-localized nodes, communication complexity, and computational complexity in realistic simulation environment. Estimation error is defined as the average variance between estimated location and real location. Residual ratio of un-localized nodes (called residual ratio for short) is the ratio of un-localized nodes after certain rounds of localization, which can be used to reflect the convergence speed as well as the localization ability of algorithm. Communication and computational complexity are defined respectively as the total number of packets exchanged for and the total amount of time spent on localization.

Network settings generally have great influence on performance of localization algorithms, among which average connectivity (AC), number of anchors (AN), and distribution of nodes (ND) are the most important ones. Average connectivity is defined as the average number of neighbors per node, and is an indicator of network density. Evenly deployed network is assumed by many localization algorithms, yet it is hard to achieve in real world. The adaptability to irregular node distribution is crucial to the robustness of algorithm. To evaluate these algorithms in a realistic simulation environment, we use RIM model to model the radio propagation pattern, and change the radio irregularity via DOI and VSP.

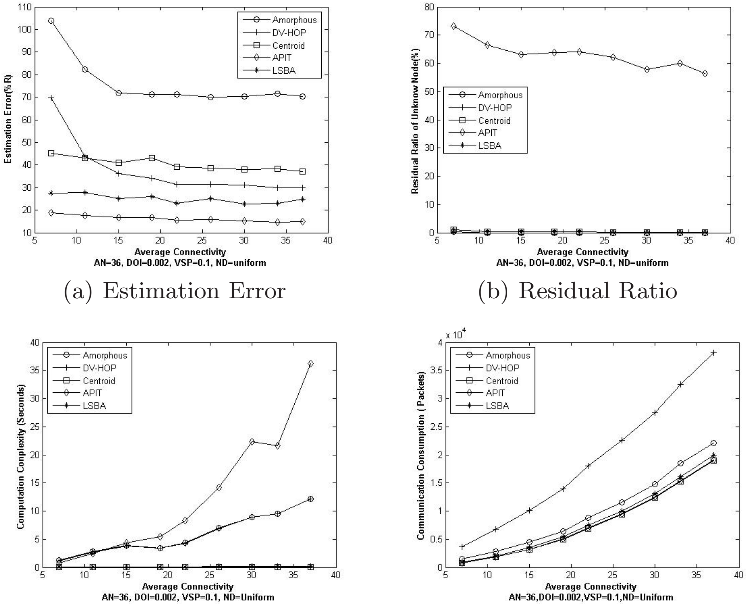
The values of DOI, VSP, Weibull and the environment noise are all from [14], which is obtained from the empirical data.

We conducted the experiments on Matlab 7.0.4 that runs on laptop IBM ThinkPad R51BC1, and omit the detailed implementation of the other algorithms due to limitation of space. For ease of comparison we didn't include iterative localization refinement in our experiments, since no such process exists in DV-HOP and Amorphous, and it greatly increases the computational complexity of APIT. For communication complexity, we didn't take collision, packet lost and traffic control into account. For computational complexity, we only counted the time spent on localization related calculations, and the computation methods are derived from APIs of Matlab without any modification.

In the following experiments, without specification, sensors (include anchors) are uniformly distributed in an area of $300 \times 300m^2$; the free space propagation radius of sensors is 48.9m; the value of DOI and VSP are 0.002 and 0.1 respectively; the value of Weibull is [0.16, 0.67]. Each experiment was run 500 times with different random seeds to get the performance.

4.1 The Influence of Average Connectivity

In this experiment, we investigate the influence of average connectivity (AC) on algorithm performance with $AN = 36$.



(a) Estimation Error

(b) Residual Ratio

(c) Computation Complexity

(d) Communication Cost

Fig. 3. The Influence of Average Connectivity

In fig.3(a), APIT has the lowest estimation error with LSBA following on the heels of it. In fig.3(b), the residual ratio of APIT is above 50% after two rounds of localization, yet the residual ratio of other algorithms almost reach 0. This is due to the stringent localizable constraints in APIT that each localizable node must reside in at least one triangle formed by three location-aware neighbors, whereas other algorithms have much relaxed constraints. Similar results are observed in other experiments indicating that APIT has the lowest convergence speed. In fig.3(c), the computation time of APIT increases most rapidly, since larger AC brings more nodes to participate in the PIT test; DV-HOP and Amorphous nearly overlap and grow quickly as well, this is because they use the same estimation methods; Centroid and LSBA have the lowest computation complexity. In fig.3(d), communication complexity of DV-HOP and Amorphous increase most rapidly and DV-HOP consumes more packets than Amorphous, since they need more packets to construct routing tables and DV-HOP has to broadcast Hop-Size whereas Amorphous computes it offline.

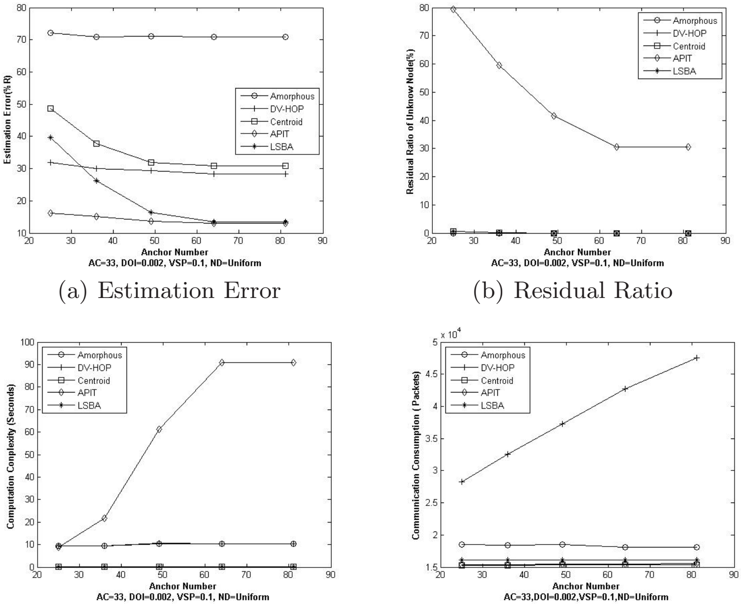
DV-HOP and Amorphous can get better estimation accuracy at higher AC, which also leads to higher communication cost; APIT achieves the best estimation accuracy, but its residual ratio and computational complexity is the highest; LSBA and Centroid have the lowest residual ratio, communication and computational complexity, and LSBA has better estimation accuracy than Centroid.

4.2 The Influence of Number of Anchors

In this experiment, we study the influence of number of anchors on algorithm performance. Since DV-HOP and Amorphous get better estimation accuracy at high AC while other algorithms are not sensitive to it, we choose to conduct this experiment with $AC = 33$.

In fig.4(a), DV-HOP, Amorphous and APIT are not sensitive to the number of anchors. Estimation error of Centroid and LSBA decrease rapidly when number of anchors increases, and LSBA performs much better than Centroid as it makes use of the link information. Although APIT achieves the highest location accuracy with only a few anchors, more anchors will greatly improve its convergence speed, see fig.4(b). In fig.4(c), the computation time of APIT increases rapidly when the number of anchors increases, since more anchors appearing in the neighborhood of an unknown node increase the number of PIT tests. Other curves hardly change, since the amount of nodes in the network doesn't change when average connectivity is fixed, and moreover the number of unknown nodes decreases when the number of anchors increases. In fig.4(d), the communication cost of DV-HOP is the highest, and moreover it increases rapidly with the number of anchors, since more anchors require more broadcast of Hop-Size.

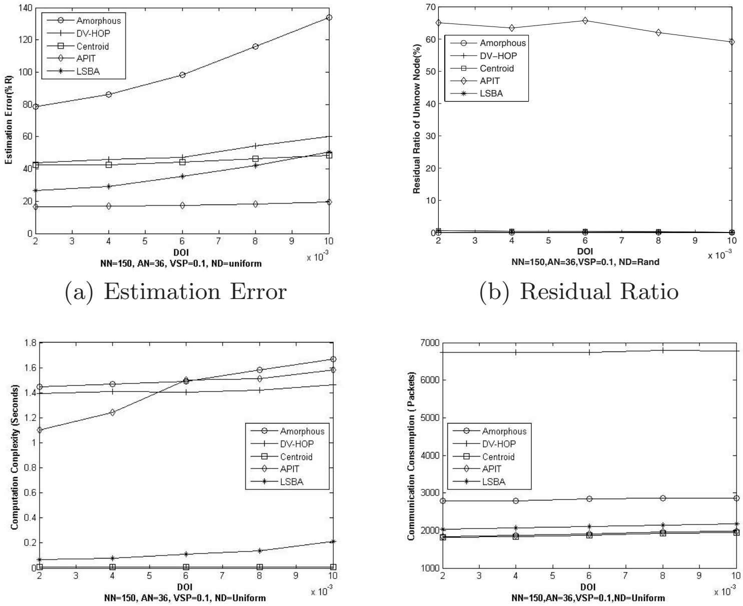
To sum up, if only a few anchors exist in the network, DV-HOP and APIT are better choices due to their lower estimation error; however when more anchors are available, LSBA is preferable as it achieves the best tradeoff between localization accuracy and energy saving.



(a) Estimation Error (b) Residual Ratio

(c) Computation Complexity (d) Communication Cost

Fig. 4. The Influence of Number of Anchors



(a) Estimation Error (b) Residual Ratio

(c) Computation Complexity (d) Communication Cost

Fig. 5. The Influence of DOI

4.3 The Influence of DOI

In this experiment, we study the influence of DOI on algorithm performance. Since average connectivity changes with DOI, we choose to fix the number of nodes (NN) instead of average connectivity in this experiment. In this case, the average connectivity is around 7 that is the most common situation in real world.

It's easy to know from formula (1) that DOI has smaller influence on RSS than VSP as it changes from 0.002 to 0.01. Nevertheless the estimation error of Amorphous is affected greatly by DOI (fig.5(a)), since its Hop-Size is calculated based on average connectivity that is influenced by DOI. The estimation error of LSBA also increases, as increased DOI enlarges the distance uncertainty. However, the estimation error of Centroid and APIT hardly change, as they only care about the locations of heard anchors, meanwhile the number of heard anchors is less affected by DOI. In fig.5(c), the computation time of DV-HOP and Amorphous is very high, because irregular signal pattern causes more inaccurate distance estimation which in turn takes the estimator more time to search the optimal solution. In fig.5(d), the communication cost of all the algorithms are not sensitive to DOI, in which DV-HOP is the highest.

4.4 The Influence of VSP

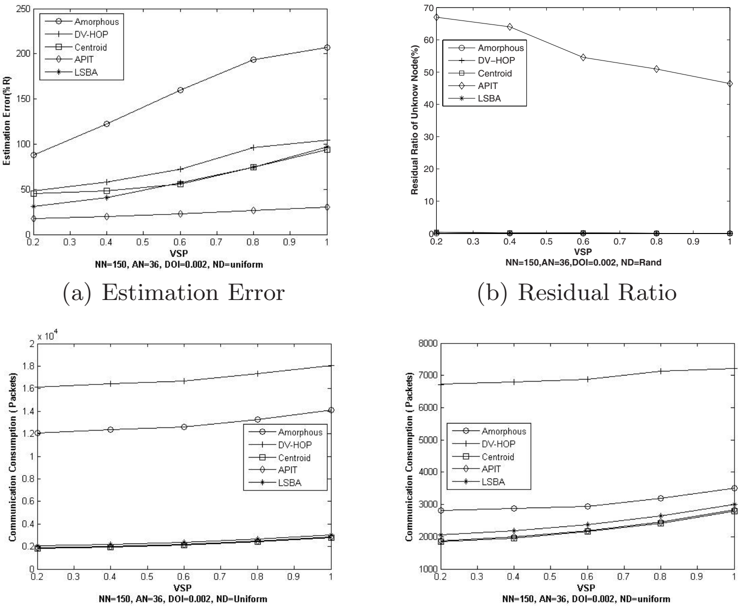
In this experiment, we study the influence of VSP on algorithm performance. Since average connectivity also changes with VSP, we fix the number of nodes instead of average connectivity in the experiment.

Compared with DOI, VSP has much greater contribution to radio irregularity when it changes from 0.2 to 1, so nearly all the algorithms experience increased estimation error in fig.6(a). An interesting observation is that increased VSP helps to reduce the residual ratio of APIT (fig.6(b)). Similar to fig.5(c), the computational complexity of DV-HOP and Amorphous in fig.6(c) is very high due to the large number of inaccurate distance estimation caused by increased VSP. Another observation in fig.6(c) is that the computation time of APIT increases very fast than that in fig.5(c), the reason may be that highly irregular radio pattern brings more nodes to the neighborhood of an unknown node, and thus increases the number of PIT tests. This also explains why increased VSP reduces the residual ratio of APIT. Similar to section 4.3, communication cost of DV-HOP is still the highest in fig.6(d).

4.5 The Convergence Speed of APIT

Above experiments show that although APIT has the highest estimation accuracy, its convergence speed is very low. In this section, we investigate the convergence characteristic of APIT using the same environment setting as that in section 4.1.

From fig.7, both the estimation error and residual ratio decrease rapidly in the first three rounds, and then decline very slowly; meanwhile the computation and communication cost increase steadily. The residual ratio is as high as 35%



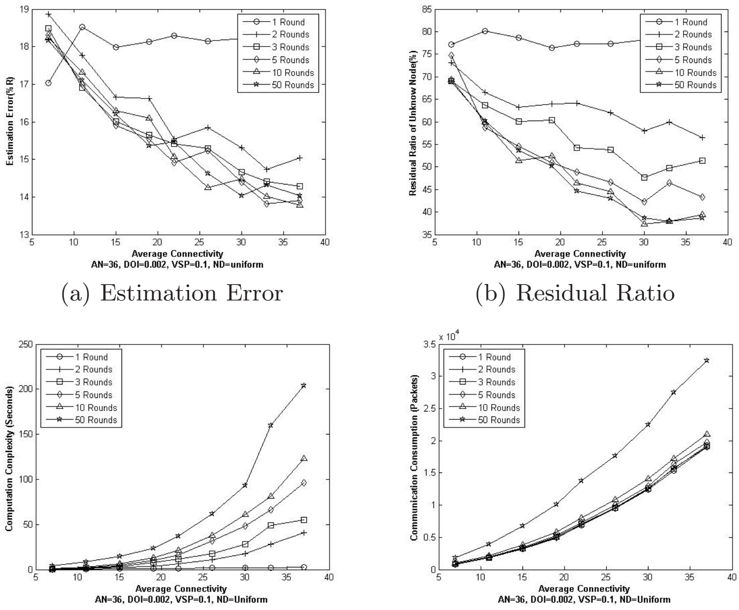
(a) Estimation Error

(b) Residual Ratio

(c) Computation Complexity

(d) Communication Cost

Fig. 6. The Influence of VSP



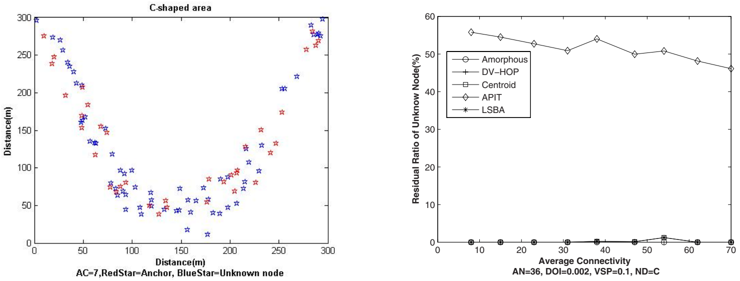
(a) Estimation Error

(b) Residual Ratio

(c) Computation Complexity

(d) Communication Cost

Fig. 7. The Convergence Speed of APIT



(a) C-Shaped Area

(b) Residual Ratio

Fig. 8. Performance in C-Shaped Area

after tens of rounds even under high AV, the reason is that the nodes near the boundary as well as those in regularly deployed area cannot find three neighbors that can hold it in a triangle. Therefore mobile anchors are needed to help APIT improve the convergence speed and localizable ratio.

4.6 Performance in C-Shaped Area

In this experiment we study the adaptivity of five algorithms to irregular node deployment. We conduct the experiment in a C-shaped area that is limited in a square with AC increasing from 7 to 70, see fig.8(a). In this section we only focus on the estimation error, since others are much similar to that in section 4.1.

In fig.8(b), the estimation error of Centroid, LSBA and APIT hardly change when AC increases, meanwhile LSBA and APIT have the lowest estimation error. The estimation error of DV-HOP and Amorphous increase quickly at first and then stay at high level afterwards. The reason is that in both algorithms accurate estimation of Hop-Size depends on both high connectivity and uniformly deployment of nodes, yet the last condition cannot be met in C-shaped area. Moreover, more participant nodes may add more estimated errors in C-shaped area, that's why DV-HOP and Amorphous suffer from high estimation error when average connectivity increases. It seems that DV-HOP and Amorphous are not suitable to work in highly irregular regions.

5 Conclusion

In this paper, we provide the motivation and implementation of LSBA, a coarse-grained RSS-based localization algorithm, and the comprehensive performance evaluation of LSBA and four other typical coarse-grained algorithms: Centroid, APIT, DVHOP and Amorphous. Simulation results show that LSBA achieves the best cost-performance ratio in WSNs with moderate number of anchors; moreover LSBA has better adaptability to irregular node deployments.

References

1. Niculescu, D., Nath, B.: Ad Hoc Positioning System. In: Proceedings of the IEEE Global Communications Conference (November 2001)
2. Bulusu, N., et al.: GPS-less Low Cost Outdoor Localization for Very Small Devices. IEEE Personal Communications Magazine (2000)
3. He, T., et al.: Range-Free Localization Schemes for Large Scale Sensor Networks. In: Proceedings of Annual International Conference on Mobile Computing and Networking (September 2003)
4. Niculescu, D., Nath, B.: DV Based Positioning in Ad hocNetworks. Journal of Telecommunication Systems (2003)
5. Nagpal, R., et al.: Organizing a Global Coordinate System from Local Information on an Ad Hoc Sensor Network. In: Proceedings of IPSN (2003)
6. Doherty, L., et al.: Convex Position Estimation in Wireless Sensor Networks. In: Proceedings of IEEE INFOCOM (April 2001)
7. Bahl, P., Padmanabhan, V.N.: RADAR: An In-Building RF-Based User Location and Tracking System. In: Proceedings of the IEEE INFOCOM (March 2000)
8. Savvides, A., et al.: Dynamic Fine Grained Localization in Ad-Hoc Sensor Networks. In: Proceedings of 7th Annual International Conference on Mobile Computing and Networking (July 2001)
9. Chintalapudi, K., et al.: Localization Using Ranging and Sectoring. In: Proceedings of IEEE INFOCOM (March 2004)
10. Niculescu, D., et al.: Ad Hoc Positioning System Using AOA. In: Proceedings of IEEE INFOCOM (April 2003)
11. Hightower, J., et al.: SpotON: An indoor 3D Location Sensing Technology Based on RF Signal Strength. University of Washington CSE Report (2000)
12. Kusy., B., et al.: Node-density independent localization. In: Proceedings of IPSN (2006)
13. Ganesan, D., et al.: Complex Behavior at Scale: An Experimental Study of Low-Power Wireless Sensor Networks. Technical Report UCLA/CSD-TR 02-0013 (2002)
14. Zhou, G., et al.: Impact of Radio Irregularity on Wireless Sensor Networks. In: Proceedings of 2nd International Conference on Mobile Systems, Applications, and Services (June 2004)
15. Lymberopoulos, D., et al.: An Empirical Analysis of Radio Signal Strength Variability in IEEE 802.15.4 Networks using Monopole Antennas, ENALAB Technical Report, Yale University, USA
16. Rappaport, T., et al.: Wireless Communications: Principles and Practice. Prentice Hall, Englewood Cliffs (2002)
17. Li, X., Hua, B., Guo, Y.: Link State Based Annulus Localization Algorithm for Wireless Sensor Networks. In: Proceedings of the International Conference on Communications and Networking in China (2007)

Research on Clustering Strategy for Wireless Sensor Network Based on Fuzzy Theory

Wei Zhenhua, Hou Xiaodong, Zhou Hong, and Liu Chang'an

Department of Computer Science and Technology
North China Electric Power University
Bejing, China
hitwzh@163.com

Abstract. In wireless sensor network applications, effective clustering algorithm can reduce energy consumption, which can increase network scalability and lifetime. In most of traditional clustering methods, clusters do not form until their cluster heads are selected and these algorithms are usually running under the single data transmission mode. A clustering strategy for WSNs based on Fuzzy Cluster Mean (FCM) is proposed in this paper. It is a new clustering mechanism of generating the clusters first and then selecting the cluster head(CH). This strategy based on FCM has good characteristics of clustering quick, reducing energy consumption and being applied in different data transmission modes. The correctness and feasibility is validated in simulations. It is shown that energy consumption is better than the similar clustering algorithms. When different selection modes of CH are selected in this strategy, the clustering strategy based on FCM shows the best efficiency in two data transmission modes.

Keywords: Wireless Sensor Network; FCM; Topology; Clustering.

1 Introduction

Wireless Sensor Network(WSN), comprising of large quantities of low-price small sensor nodes in monitor region, is a multi-hop ,self-organization network system through wireless communication. These nodes can make best use of themselves to sensor, gather and process the data which are to sent to the Base Station(BS) by cooperation. At recent years, with the development of MEMS and communication, the decrease of computing cost and the microminiaturize of the volume of the micro-processor, Wireless Sensor network are showing wide application backgrounds, such as environment monitoring, target surveillance[1,2,3], and so on.

It is known that the capability of each sensor node is limited by constraints such as constraint supply of power, manufacturing costs, and limited package size. Consequently, each sensor node has short communication and sensing ranges, a limited amount of memory and limited computational power. Considering these limitations, the critical technologies of WSN are generally designed with energy conservation and network lifetime extension. The network topology research based on clusters is always the focus in recent years. In the cluster based network, it is easy to

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 596–604, 2007.

apply the distributed algorithm. The cluster heads take the task of data aggregation and data routing to decrease the communication mount and this prolongs the whole lifetime of network dramatically. LEACH[3] is a self-adaptation topology algorithm, and it is executed periodically, that is the cluster restructure in each round. In the process of executing LEACH, some sensor nodes are randomly selected as the Cluster Heads(CHs) firstly. The other sensors choose the nearest CH as its own CH, whether or not, based on the strength of signals.

At present, many of cluster-based algorithm mechanism are with the foundation of determination of CHs firstly, and then build the cluster topology. In this paper, we propose a clustering strategy based on Fuzzy Cluster Mean (FCM), which is quite different from the traditional clustering method for WSN. In this mechanism, clusters are fixed firstly, and then in each cluster CH is selected individually for the purpose of providing fast clustering and energy conservation. At meanwhile, it is suitable to different data transmission model.

The rest of the paper is organized as follows. In Section 1, we present some background information about FCM algorithm. In Section 2, we give the network model and radio power model. In the next Section, we propose a whole solution of clustering based on FCM algorithm and in Section 4 we lay out the simulation curve and analyze the performance. At last, conclusion is given.

2 Definition of FCM Algorithm

Fuzzy Clustering Mean Algorithm is widely used in clustering process. The conception of FCM is that the objects in same cluster own the largest similarity while objects in different clusters have the smallest similarity. Let *c* be an integer which represents the number of clusters with $2 \le c \le n$, where *n* is the number of data points and x_i is the k^{th} data sample. The aim function $J(U, c_1, ..., c_c)$ defined as follows:

$$J(U, c_1, ..., c_c) = \sum_{i=1}^{c} J_i = \sum_{i=1}^{c} \sum_{j}^{n} u_{ij}^m d_{ij}^2 \quad . \tag{1}$$

Here, u_{ij} is a number between 0 and 1, c_i is the center of fuzzy cluster *i* and $d_{ij}=||c_i-x_j||$ is the Euclidean, while *m* is a real number belong to $m \in [1, \infty)$ called the fuzzy constant.

The conditions that make the sum of the square errors J the minimum in equation (1) are given in the following equation (2) and (3):

$$c_i = \sum_{j=1}^{n} u_{ij}^m x_j \bigg/ \sum_{j=1}^{n} u_{ij}^m \quad . \tag{2}$$

$$u_{ij} = 1 \bigg/ \sum_{k=1}^{c} \left(\frac{d_{ij}}{d_{kj}} \right)^{2/(m-1)} \quad . \tag{3}$$

From the above two necessary condition, FCM algorithm is a simple process of iteration. Starting this algorithm requires two parameters: one is cluster number and the other is *m*. *m*, the fuzzy constant, is configured to different values based on different applications. So, here, we take little consideration of it. Therefore, it is important to fix the number of clusters *c* if a group of sample data wants to be clustered using FCM algorithm.

3 Wireless Sensor Network Model

Before proposing our mechanism, we sum up cluster-based wireless sensor network model. Let *N* be the number of sensor nodes $\left(N_1, N_2, \ldots, N_n\right)$, assuming that they are randomly deployed in a $M \times M$ area. Each node has the same hardware facility, sensor parts and gathering radio. The Base Station is located in the center of the network and it can get the tow-dimension coordinator of sensor node in the whole network through flooding query. All nodes are standstill.

In WSN, communication among nodes needs to consume energy while the energy resources are so limited for the sensors that they can not be supplied continuously, so this issue directly determines the lifetime of WSN. In this paper, the proposed algorithm aims at energy conservation and we use a radio model[1] in which the energy dissipation $E_T(k,d)$ of transmitting k-bit data between two nodes separated by a distance of d meters is given as follows:

$$E_T\left(k,d\right) = k(E_{elec} + \varepsilon_{amp} \cdot d^2) \ . \tag{4}$$

The power consumption for receiving sensor:

$$E_R(k,d) = E_{elec} \cdot k \ . \tag{5}$$

Where E_{elec} denotes electric energy and ε_{amp} denotes transmit amplifier parameters, which are constants as previously defined. From the above equation, the power consumption is a second order function of distance, so multi-hop communication as far as possible can get more energy efficiency than single-hop communication.

4 Clustering Mechanism Based on FCM for Wireless Sensor Network

Upon the analysis of FCM, it's easy to know that the prerequisite condition to cluster using FCM algorithm is to fix the value *c*, the number of the clusters, in n data samples. Firstly, basing on the wireless communication information theory and combining the WSN radio power model and the accomplishment in literature [4][5], we give the process of the determination of *c* given the n nodes in our clustering strategy. And then according to the location of all nodes and the cluster number *c*, we can get the *c* Cluster Heads based on the two-dimension coordinator and a fuzzy cluster matrix U through FCM algorithm. Therefore, clustering process to the whole network is completed.

4.1 Related Information Theory

The basic idea of Distributed Source Coding(DSC)[7] is to use redundant information between the sensors to reduce final data size for transition. *H(N$_i$)* denotes the node *N$_i$*'s information . Based on the DSC, the information obtained by a whole network can be evaluated by the entropy *H(N$_1$,N$_2$,...,N$_n$)*. This value should be bigger or the same as the sum of information obtained by single node. In the cluster-based Wireless Sensor Network, the information that is only provided by the node Ni can be expressed by entropy:

$$H(N)-H(N\cap\overline{N_i}) \ . \tag{6}$$

The coefficient p_i is defined as the percentage of unique information of sensor N_i compared to $H(N_i)$, the complete information provided by N_i:

$$p_i=\frac{H(N)-H(N\cap\overline{N_i})}{H(N_i)}(0<p_i\le 1) \ . \tag{7}$$

The coefficient p_i can express the degree of correlation between a sensor and its neighborhood sensors.

4.2 Dividing the Network into Clusters

We consider *C=N/s* clusters each consisting of *s* sensors. The total number of bits-hop cost for the whole sensor network is expressed as:

$$E_{whole}=\sum_{i=1}^{c}(E_{\text{int}ra}(i)+E_{extra}(i))=c(E_{\text{int}ra}+E_{extra}) \tag{8}$$

Where $E_{\text{int}ra}(i)$ and $E_{extra}(i)$ are the bit-hop cost within cluster i and the bit-hop cost for cluster i to the sink respectively. And the average bit-hop cost in the above two conditions and we can obtain expressions for each of these:

$$\begin{aligned}E_{\text{int}ra}&\propto((s-1)Hp)(d\sqrt{s})\\E_{extra}&\propto(H+(s-1)Hp)d\sqrt{N}\end{aligned} \ . \tag{9}$$

Where *H*、*p and d* is the average of $H(N_i)$、 p_i and d_i. $(s-1)Hp$ and $d\sqrt{s}$ are the average of bits of all sensors except the head of cluster in a cluster, and average distance from the sensors to the head of cluster respectively. $H+(s-1)Hp$ and $d\sqrt{N}$ are average number of bits of the head of cluster after data fusion and average distance from the head of cluster to the root. The number of sensors in each cluster is much larger than one. So, we have :

$$\begin{aligned} E_{whole} = KE_{\text{int}ra} + KE_{extra} \propto \sqrt{s}HpdN + \\ s^{-1}H(1-p)dN\sqrt{N} + HpdN\sqrt{N} \end{aligned} \quad . \tag{10}$$

The optimum value of the cluster size can be determined by setting the derivative of the above expression equal to zero:

$$\frac{\partial E_{whole}}{\partial_s} = 0 \Rightarrow \frac{p}{2\sqrt{s}} - s^{-2}(1-p)\sqrt{N} = 0 \ . \tag{11}$$

The optimum number of clusters can be expressed as:

$$C_{opt} = \frac{N}{s_{opt}} = (\frac{pN}{2(1-p)})^{\frac{2}{3}} \ . \tag{12}$$

From the equation (12), we can see that the optimum number of clusters depends on the number of sensor in the entire sensor network *N* and the degree of correlation p.

4.3 Cluster-Based Algorithm

From the optimum cluster number *c* given by above section and taking the two-dimension coordinators of n sensor nodes as sample data, we divide the *n* coordinator $\{(x_1,y_1),(x_2,y_2),..(x_n,y_n)\}$ into clusters by FCM method, where represents the real distance from the centre of i^{th} cluster to j^{th} node. The detailed steps are as follows:

(1) Choose random real number between 0 and 1 to initialize the node association matrix;
(2) Get *c* cluster centres C_i using equation 2, *i=1,...,c.*
(3) Compute the aim cost function based on equation 1. If the result is smaller than certain threshold, or the change amount relative to the previous cost function result is smaller than certain threshold, stop the procedure.
(4) Compute the association matrix *U* using equation 3. Return to the step 2.

The algorithm above also can be performed under the situation that initialize the cluster centre first and then go on the iteration process. The output of algorithm is *c* cluster centre coordinator and a *c*×*n* node association matrix which represents the cluster correlation of each sample point. Basing on this output matrix and according to the biggest association rules in fuzzy collections, we can fix the each coordinator's membership of some cluster. Cluster centre point denotes the average characteristic of a cluster and they can be thought to be the representative of each cluster.

4.4 Transmitting the Cluster-Formation Message

The cluster dividing strategy based on FCM is a simple process of iteration As the procedure is performed in the Base Station, we needn't take much consideration into the energy consumption produced by the complexity of algorithm. After the cluster dividing, in order to let all nodes know which cluster they belong to, the base station need to broadcast the cluster information to the whole network. Upon the output matrix *U* produced by FCM algorithm, the Base Station provides *c* different cluster packages and then broadcast them one by one. The i^{th} message includes ClusterID, the center of Ci^{th} cluster and all coordinators in this cluster. The behavior of node on receiving messages is described in Algorithm 1.

Algorithm 1. Cluster Algorithm in This Paper

Input：message[i] Output：FALSE,TRUE

```
if(message[i] arrived)           //Message arrived
 if(node.ClusterCenter)
              //Deciding whether the node belongs some cluster
      {discardMessage();
               /*If not NULL, it denotes that the node has fixed the
                      cluster which it belongs to, and then desert the
                      message.*/
             return FALSE;
      }
   else{
      if(find(message[i],node.coordnt)){
             /*If ClusterCenter field is NULL, search in the message
             whether there is the node coordinate in it;if found,
             find() returns TRUE.*/
        node.ClusterCenter=getCenter(message[i]);
               /*The node get the ClusterCenter field from the message
             and then fixed which cluster it belongs to*/
        node.ClusterI=getID(message[i]);
               //Fix the ClusterID
        return TRUE;
     }
     else{
        discardMessage();
             /*If there is no value in the message to match the
             coordinate of the node, desert the message.*/
        return FALSE;
      }
  }
```

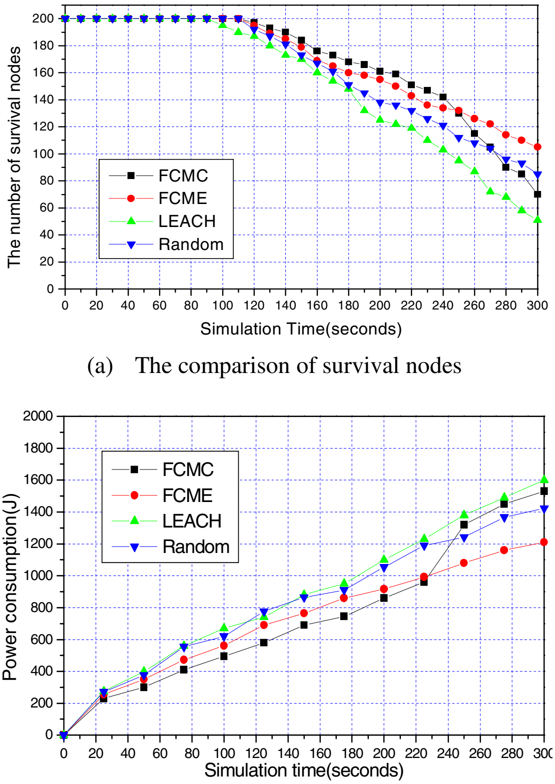
4.5 Selecting the Heads of Clusters

In Wireless Sensor Network, information provided by sensor nodes has two types: continuous mode and discontinuous mode. Accordingly, the transmission mode in WSN can be divided into continuous data stream mode and event-driven mode. The application precondition to the former routing protocol, based on clusters, is that each node is in active mode, continuously performs data gathering, processing, sending and receiving and periodically rotates its Cluster Head in avoidance of the early failure of some node. In event-driven routing, the node in the monitor area will be inactive until the events occur and in the situation data transmission will happen. Many parts of nodes are normally in the sleeping or shut-down state except of some components in active state in order to save energy.

After the cluster topology process using FCM, as long as choosing the proper CH selection mechanism, we can complete the cluster topology establishment of the whole network. We will take three selection methods, including cluster-centre based, remaining-energy based and random-selection, in continuous data stream mode and event-driven mode separately. The simulation in the following section indicates the application scope while taking the above measures after carrying out our clustering strategy based on FCM. The clustering topology of the whole network is completed the cluster dividing and CH selection.

5 Simulation and Performance Evaluation

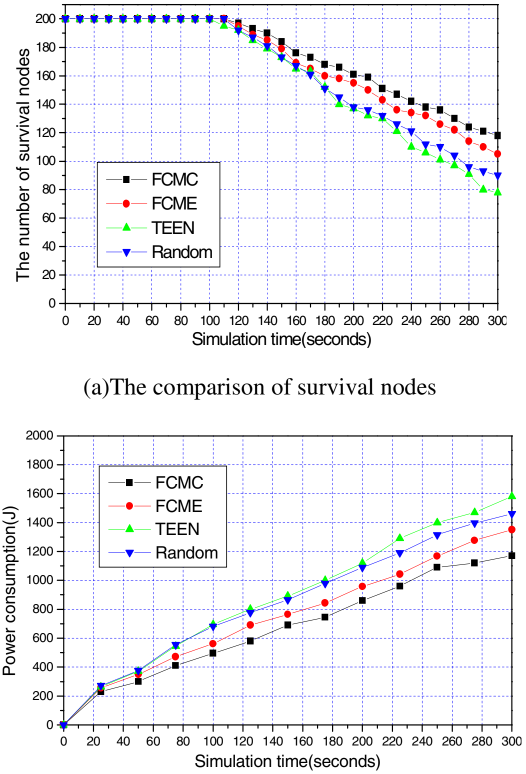
200 nodes were initially positioned at random locations over 150*150m^2 area. Node sensing range, wireless communication range and the sensing ratio interval are separately 12m, 50m and 0.06s. Each node has the initial energy 10 Joules. Every experiment is taken with 300 seconds. After many tests, we get some important parameters simulation curve under the continuous data stream mode and event-driven mode separately in Figure 1 and 2, where FCMC represents the cluster-centre based strategy, FCME represents remaining-energy based strategy and Random represents random-selection strategy. The three methods above are all taken after our cluster dividing mechanism based on FCM.



(a) The comparison of survival nodes

(b) The comparison of energy consuming

Fig. 1. Performance comparison under continuous data flow transmission mode



(a)The comparison of survival nodes

(b)The comparison of energy consuming

Fig. 2. Performance comparison under event-driven transmission mode

Figure 1 denotes the curve of the change of remaining nodes and energy-consumption under the continuous data stream mode. From the figure, we can see that, before the 205th second, FCMC shows the better performance, but after 205th second, because of the continuous monitor, failure of some CHs appears .This cause the topology detachment of some cluster and the whole network and the number of failure nodes increases dramatically (the energy consumption of node detached with the topology is defined to be zero).From the Figure, under the continuous data stream mode, FCMC shows the more instability than FCME and Random strategies. FCME indicates the best performance. Figure 2 denotes the curve of the change of remaining nodes and energy-consumption under event-driven mode. We assume nodes are periodically in working state within 5 seconds and next, in sleeping state within 5 seconds too. It’s easy to see that every CH-selection methods are all stable, but FCMC got the less energy consumption and more remaining node. Upon the above analysis, FCMC indicates the optimum performance in event-driven mode.

6 Conclusion

In this paper, we present a cluster-dividing method based on FCM algorithm. It changes the traditional way of clustering and adopts the clustering firstly and then CH-selection mechanism. The cluster strategy given by this paper has merits with rapid and steady cluster process and energy conservation. Meantime, simulation indicates that when adopting FCME, the performance of our strategy under continuous data stream mode is the best and when adopting FCMC, we can see that our strategy under event-driven mode shows the best performance.

References

1. Guangyang, H., Xiaowei, L.: Energy-efficiency Analysis of Cluster-based Routing Protocol in Wireless Sensor Network. IEEE Aerospace Conference(2006)
2. Ossama, Y., Marwan, K., Srinivasan, R.: Node Clustering in Wireless Sensor Networ:Recent Developments and Deployment Challenges. IEEE Network 20, 20--25(2006)
3. Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H.: An Application-specific Protocol Architecture for Wireless Microsensor Networks. IEEE Transactions on Wireless Communications, pp.660--670(2002)
4. Limin, S., Jianzhong, L., Yu, C., et.al.: Wireless Sensor Network. Tsinghua University Publication, Beijing(2004)
5. Heinzelman, W.R., Kulik, J., Balakrishnan, H.: Negotiation-based Protocol for Disseminating Information in Wireless Sensor Network. Wireless Networks 8, 169--185(2002)
6. Lin, F., Huanzhao, W., Hai, W.: A Solution of Multi-target Tracking Based on FCM Algorithm in WSN. In: 4th Annual IEEE International Conference on Pervasive Computing and Communications Workshops, pp. 124-128(2006)
7. Zixiang, X., et.al.: Distributed Source Coding for Sensor Networks. IEEE Signal Processing Magazine 9: 72-76(2004)

Performance Analysis of the Energy Fairness Cooperation Enforcement Mechanism (EFCEM) in Ad Hoc Networks*

Wu Hao, Ding Yi-ming, and Li Cheng-shu

State Key Laboratory of Rail Traffic Control and Safety
Beijing Jiaotong University, Beijing, P.R. China 100044
{hwu,06120165,csli}@bjtu.edu.cn

Abstract. Mobile ad hoc networks must operate independent of fixed or centralized network management infrastructure, the communication between nodes rely on the node's cooperation, so nodes' selfish behavior to save the battery cost for their own communication cannot be neglect. However, general cooperation enforcement mechanisms do not consider node battery, so hotspot node issue may exist; even badly lead to network separation. This paper proposes an Energy Fairness Cooperation Enforcement Mechanism (EFCEM) in ad hoc networks. Theoretical analysis and numerical simulation results show that the EFCEM protocol, comparing with the traditional reputation mechanisms, can prolong networks existence time and improve network throughput, but at the cost of a little more packets delay.

Keywords: Ad hoc networks, Energy fairness, and Cooperation enhancement mechanism.

1 Introduction

Mobile ad hoc networks must operate independent of fixed or centralized network management infrastructure, the communication between nodes rely on the node's cooperation. Because mobile nodes usually use battery to supply power, it leads to peculiar selfish behavior which means that node lack cooperation in order to save battery for own communication, i.e. node doesn't route and forward packets for others. Selfish behavior can do harm to the performance of ad hoc networks badly [1], so it's necessary to use some cooperation enforcement mechanisms to enforce node cooperation. Researchers have pay attention to node selfish behavior issues and proposed some solutions [3-6], such as reputation cooperation mechanisms that are more appropriate to solve the selfish issues [5-6].

Cooperation enforcement mechanisms are tied up with routing algorithm. When nodes in ad hoc networks forward packets, no matter which routing protocol is adopted, generally speaking, the criterion is fewest hops. Under the circumstances

[*] This paper is supported by the National Natural Science Foundation of China under Grant No.60372093; the BJTU Foundation under Grant No.2004SM003.

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 605–615, 2007.

some nodes in ad hoc networks will be a hotspot location. Hotspot nodes have to forward more packets than other nodes, so their battery consumption is quicker. If some or other hotspot's battery is out, it may lead to network partition, even network paralysis, here every node can't communicate with each other normally. Under the circumstances hotspot node is confronted with two choices: one is becoming "selfish", and the other is battery out very quickly. Neither choice is good for fairness cooperation.

This paper proposes an Energy Fairness Cooperation Enforcement Mechanism (EFCEM) in ad hoc networks. It is obvious that the EFCEM, which considering node battery is more fairness and prolongs the network existing time. A more detailed description of the EFCEM protocol is presented in section 2. Section 3 gives the performance analysis of EFCEM, and the numerical simulation results of EFCEM protocol comparing with the traditional reputation mechanisms performance is shown in section 4. Section 5 concludes this paper.

2 Energy Fairness Cooperation Enforcement Mechanism

The Energy Fairness Cooperation Enforcement Mechanism (EFCEM) is based on the reputation mechanisms. It need monitor neighbor nodes and compute the reputations of neighbor nodes, aim to eliminating selfish nodes by not using selfish nodes when routing or forwarding and defying service for selfish nodes. Meanwhile, it also considers to protect the battery cost of hotspot nodes [7].

2.1 Confirmation of Neighbor Node's Active Time in the Networks

In order to confirm the neighbor node's active time in the networks, every node broadcasts HELLO messages periodically, in the same time offers connection messages with neighbor node. The subsistence time of HELLO (*TTL*) is supposed equal to 1, then spread of HELLO limits to send node and neighbor node.

2.2 Reputation of Neighbor Node

Reputation mechanisms have two functions: one is punishing selfish nodes and enforcing cooperation, the other is offering "reputation" for the credit mode[5-6]. If calculated node reputation is lower than the threshold θ_1 ($0<\theta_1<1$), then this node will be judged a selfish node. Threshold θ_1 is preset and dynamic on the basis of network settings.

2.3 The Credit Mode

The credit mode is used for judging whether node is trusty or not[8]. When neighbor node that claims a low battery is judged trusty, it will be protected on routing.

The credit mode can be described as Neighbor node (reputation, time in network, historical time in network). A neighbor node that claims a low battery will be judged trusty if it meets following three conditions:1) reputation value>θ_1, time in network>T_2;2) reputation value>θ_2, time in network> T_1;3) for a new joined node, reputation value>θ_3, time in network< T_1, historical time in network> T_3;The chosen of parameters should reach some demands: $0<\theta_1<\theta_2<\theta_3\leq 1$, $0< T_1< T_2$.

On the assumptions that hotspot node is P, H is one neighbor of P, and P is not a new joined node. If at a time the data about P from H's credit mode is that reputation θ and time in network T, then the judge process is:

```
if $\theta<\theta_1$   then  $P$ is selfish and untrusty
else  ( if $\theta<\theta_2$
        ( if $T>T_2$  then  $P$ is trusty
          else  $P$ is untrusty
         )
      else ( if $T>T_1$  then  $P$ is trusty
               else  $P$ is untrusty
              )
      )
```

2.4 The Improvement on DSR Routing Protocol

Basic scheme is like that: the node which has consumed more battery is protected in route by neighbor node which tries its best not to forward those packets whose destination is not the hotspot node, and it demands some improvement on present DSR routing protocol.

Supposed the network topology as Fig. 1, *S* is the source node, *D* is the destination node, and P is a hotspot node. According to the hops, *S-H-P-D* is the best route, but due to the *P* is a hotspot node, the packages from other nodes, such as *C*,*E*, will be forwarded by *P* too. So, the power of *P* may be exhausted soon, and then the network may be divided into several parts (as shown in Fig. 2), the nodes cannot communication with each other normally. Therefore, its neighbor nodes should protect the hotspot node *P*, it may choose the route *S-H-A-B-D* to avoid forwarded by node *P*.

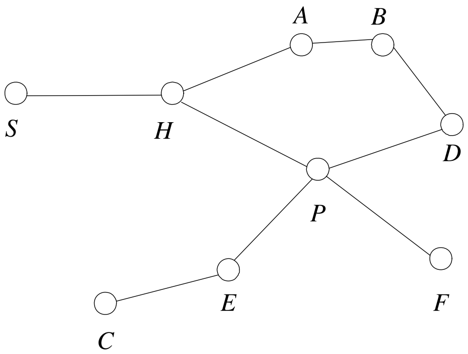


Fig. 1. Original network topology

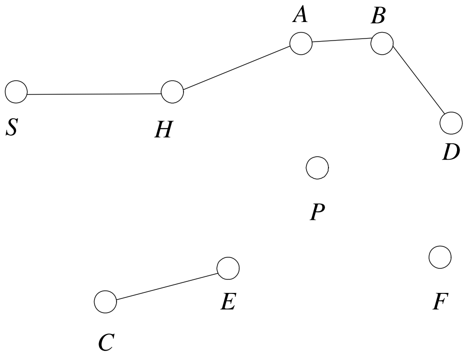


Fig. 2. Network topology without hotspot *P*

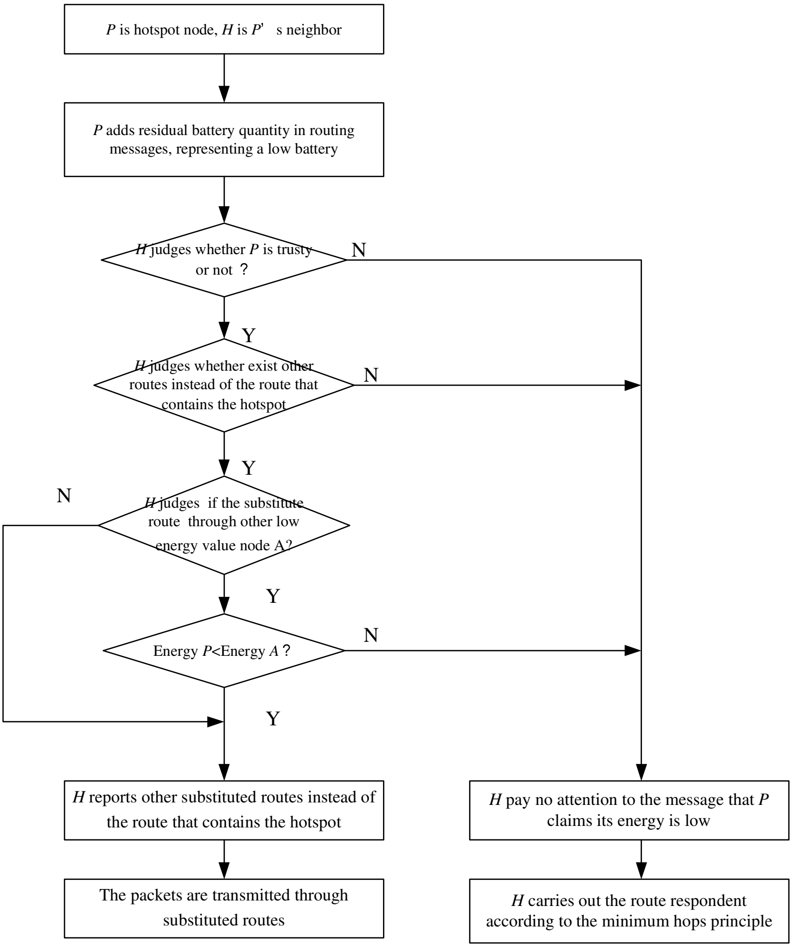

P is hotspot node, *H* is *P*' s neighbor
P adds residual battery quantity in routing messages, representing a low battery
H judges whether *P* is trusty or not ？
N
Y
H judges whether exist other routes instead of the route that contains the hotspot
N
Y
N
H judges if the substitute route through other low energy value node A?
Y
Energy *P*<Energy *A* ?
N
Y
H reports other substituted routes instead of the route that contains the hotspot
The packets are transmitted through substituted routes
H pay no attention to the message that *P* claims its energy is low
H carries out the route respondent according to the minimum hops principle

Fig. 3. Trusty mechanism and the protection to hotspot node

Each node will maintain a table that contains the trusty nodes but with a low battery among its neighbors. Suppose that *P* is the hotspot node with a low battery, and *H* is the neighbor which protects *P*, the flow chart is shown in Fig.3, then we can conclude the steps of protecting hotspot as follows:

1) *P* adds residual battery quantity in routing messages, representing a low battery;
2) *H* inquires if there are other substituted routes to the destination;
3) *H* judges whether *P* is trusty or not by the credit mode;
4) *H* reports other substituted routes instead of the route that contains the hotspot;
5) The packets are transmitted through substituted routes;
6) Hotspot node alleviates battery consumption.

If node battery is considered in cooperation enforcement mechanism, the EFCEM using in DSR routing protocol can be summarized as follows:

1) When the battery is low, hotspot adds residual battery quantity in routing messages to neighbor nodes for protection, but whether protected or not is judged by the credit mode. Protect trusty nodes, but don't protect untrusty nodes.

2) If there is only one routing to destination, then regardless of trusty or untrusty nodes, it will not be protected.

3) When there are two or more nodes claiming low batteries and all are trusty, if there exist other substituted routes, then choose the route where there aren't low battery nodes; if not, then choose the route where the battery of node with a low battery is the most comparatively.

3 Performance Analysis

An ad hoc network describes as follows: n is the number of nodes; J_0 is the initialization of every node battery; j is the battery consumption that a node sends or forward a packet; r is the number of packets which every node sends as the source in unit time; m is the average length of packets, which unit is bit; v bit per unit time stands for packets transmission velocity between nodes; d units time stands for packets forwarding delay in the intermediary nodes; other battery consumptions except sending and forwarding packets, and the packets receiving delay are ignored.

On the assumptions that in the time of t units b routes through which packets are transmitting exist; average route hops from source to destination node is h; only one node locates hotspot, and among these b routes there are $a(0<a<b)$ routes that need hotspot node's forwarding according as general minimum hops routing protocol, and there are c $(1 \le c \le a)$ nodes share in forwarding packets for hotspot node, substituted route hops is $h+s$ $(s>0)$.

3.1 Performance Analysis of the Reputation Mechanisms Not Considering Battery

In ad hoc networks with the reputation mechanisms not considering battery, the network nodes can be classified as two kinds: one kind is hotspot node; another kind is non-hotspot node.

The average transmission time between two nodes that may communicate with each other directly is $t_1 = \frac{m}{v}$ (unit time), one node cost $t_2 = \frac{1}{r}$ (unit time) to generates a packet. A packet average transmission time from source node to destination node is $t_3 = t_2 + h \cdot t_1 + (h-1) \cdot d$ (unit time). Within *t* unit time, the correct received packages number is $k_1 = [\frac{t - t_3}{t_2}]$, the un-received packages (still on the routing) number is $k_2 = [\frac{t_3}{t_2}]$, where [*] stands for maximum integer that is not larger than *. To simply the calculation, we suppose that the average hops number of those packages un-received by destination node is $\frac{h}{2}$ during the *t* unit time, energy consuming is $\frac{h}{2} \cdot j \cdot k_2$, the *n* nodes in the network share the energy consuming in the process averagely.

In ad hoc networks with the reputation mechanisms not considering battery, networks exist time *T*, networks throughput S_t in the time of *t* units, average packets transmission delay t_d and networks overall throughput S_T in the time of *T* is given by:

$$T = \frac{2n \cdot J_0 \cdot t}{j} \cdot \frac{1}{2n \cdot t \cdot r + 2n \cdot a \cdot g_1 + h \cdot g_2} \tag{1}$$

$$S_t = b \cdot g_1 \tag{2}$$

$$t_d = \frac{1}{r} + \frac{h \cdot m}{v} + (h-1) \cdot d \tag{3}$$

$$S_T = \frac{2n \cdot J_0}{j} \cdot \frac{b \cdot g_1}{2n \cdot t \cdot r + 2n \cdot a \cdot g_1 + h \cdot g_2} \tag{4}$$

$$g_1 = \left[r \cdot t - 1 - \frac{r \cdot h \cdot m}{v} - r \cdot (h-1) \cdot d \right] \tag{5}$$

$$g_2 = \left[1 + \frac{r \cdot h \cdot m}{v} + r \cdot (h-1) \cdot d \right] \tag{6}$$

where [*] stands for maximum integer that is not larger than *.

3.2 Performance Analysis of the EFCEM

In ad hoc networks with the EFCEM, the network nodes can be classified as three kinds: hotspot node, substitute node and normal node. Since the average hops of substitute route is more than general route, the package transmission delay will be discussed respectively.

A packet average transmission time from source node to destination node through the substitute route is $t_{31}=t_2+(h+s)\cdot t_1+(h+s-1)\cdot d$ (unit time). Within *t* unit time, the correct received packages number is $k_{11}=[\frac{t-t_{31}}{t_2}]$, the un-received packages (still on the routing) number is $k_{21}=[\frac{t_{31}}{t_2}]$, where [*] stands for maximum integer that is not larger than *. To simply the calculation, we suppose that the average hops number of those packages un-received by destination node is $\frac{h+s}{2}$ during the *t* unit time, energy consuming is $\frac{h+s}{2}\cdot j\cdot k_{21}$.

A packet average transmission time from source node to destination node through the normal route is $t_{32}=t_2+h\cdot t_1+(h-1)\cdot d$ (unit time). Within *t* unit time, the correct received packages number is $k_{12}=[\frac{t-t_{32}}{t_2}]$, the un-received packages (still on the routing) number is $k_{22}=[\frac{t_{32}}{t_2}]$, where [*] stands for maximum integer that is not larger than *. To simply the calculation, we suppose that the average hops number of those packages un-received by destination node is $\frac{h}{2}$ during the *t* unit time, energy consuming is $\frac{h}{2}\cdot j\cdot k_{22}$.

So the average transmission time of one package from source node to destination node is $t_3=\frac{k_{11}\cdot t_{31}+k_{12}\cdot t_{32}}{k_{11}+k_{12}}$, the whole energy consuming of those packages that un-received by destination node during *t* unit time is $\frac{h+s}{2}\cdot j\cdot k_{21}+\frac{h}{2}\cdot j\cdot k_{22}$,the *n* nodes in the network share the energy consuming in the process averagely.

In ad hoc networks with the EFCEM, networks exist time *T'*, networks throughput S_t*'* in the time of *t* units, average packets transmission delay t_d*'* and networks overall throughput S_T*'* in the time of *T'* is given by:

$$T^{'} = \begin{cases} \dfrac{2n \cdot J_0 \cdot t}{j} \cdot \dfrac{1}{2n \cdot t \cdot r + 2n(a-c) \cdot g_1 + (h+s)g_4 + h \cdot g_2} & 0 < \dfrac{c}{a} < k_0 \\ \dfrac{2n \cdot J_0 \cdot t}{j} \cdot \dfrac{1}{2n \cdot t \cdot r + 2n \cdot c \cdot g_3 + (h+s)g_4 + h \cdot g_2} & k_0 < \dfrac{c}{a} < 1 \end{cases} \tag{7}$$

$$S_t' = c \cdot g_3 + (b-c) \cdot g_1 \tag{8}$$

$$t_d^{'} = \frac{g_3 \cdot (\frac{1}{r} + (h+s) \cdot \frac{m}{v} + (h+s-1) \cdot d) + g_1 \cdot (\frac{1}{r} + h \cdot \frac{m}{v} + (h-1) \cdot d)}{g_3 + g_1} \tag{9}$$

$$S_T^{'} = \begin{cases} \dfrac{2n \cdot J_0}{j} \cdot \dfrac{c \cdot g_3 + (b-c) \cdot g_1}{2n \cdot t \cdot r + 2n(a-c) \cdot g_1 + (h+s)g_4 + h \cdot g_2} & 0 < \dfrac{c}{a} < k_0 \\ \dfrac{2n \cdot J_0}{j} \cdot \dfrac{c \cdot g_3 + (b-c) \cdot g_1}{2n \cdot t \cdot r + 2n \cdot c \cdot g_3 + (h+s)g_4 + h \cdot g_2} & k_0 < \dfrac{c}{a} < 1 \end{cases} \tag{10}$$

(where k_0 is a given numerical value)

$$g_3 = [r \cdot t - 1 - r \cdot (h+s) \cdot \frac{m}{v} - r \cdot (h+s-1) \cdot d] \tag{11}$$

$$g_4 = [1 + r \cdot (h+s) \cdot \frac{m}{v} + r \cdot (h+s-1) \cdot d] \tag{12}$$

4 Simulation and Result Analysis

Compared results are $T<T'$, $S_t>S_t'$, $t_d<t_d'$, $S_T<S_T'$,and simulation results are shown in Fig.3-Fig.6(where, the broken lines stand for the reputation mechanisms not considering battery, and the real lines stand for the EFCEM considering battery).

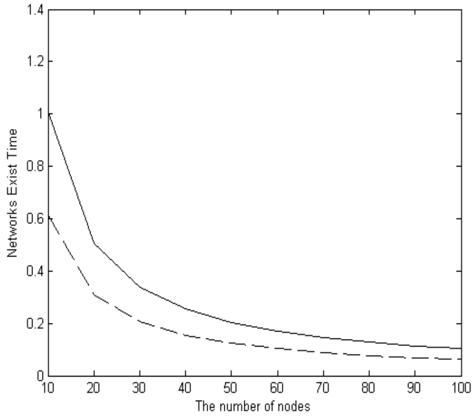


Fig. 4. Network Exist Time

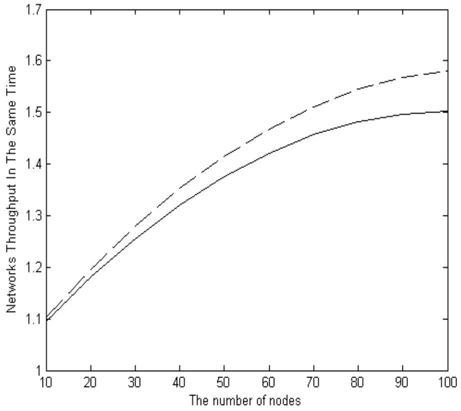


Fig. 5. Networks Throughput in the Same Time

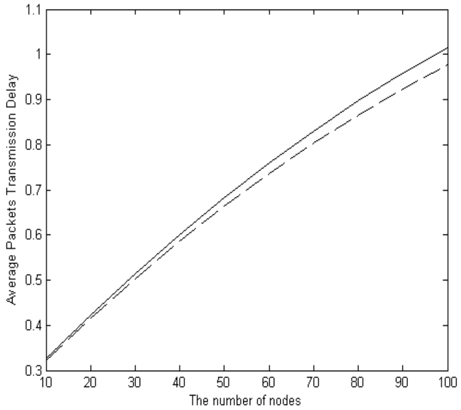


Fig. 6. Average Packets Transmission Delay

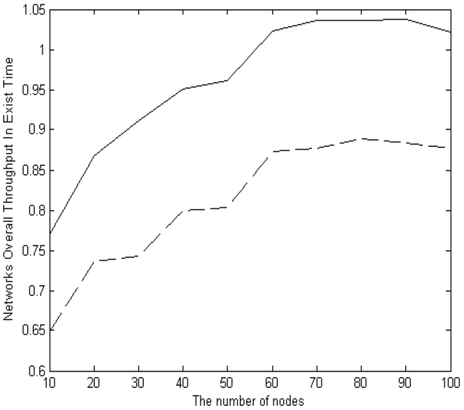


Fig. 7. Network Overall Throughput in Exist Time

Above comparison results are consistent with aforehand analysis and practical cases, and analysis describes as follows:

(1) Networks Life Time
The EFCEM protects hotspot node effectively, alleviates battery consumption, and transfers hotspot battery consumption to non-hotspot partly, so networks life time is longer.

(2) Average Packets Transmission Delay
Tue to routes that the EFCEM has chosen are not optimal, as fewest hops as possible, average packets transmission hops of the EFCEM is larger than that of the reputation mechanisms not considering battery. As a result, average packets transmission delay of the EFCEM is a little longer than that of the reputation mechanisms not considering battery. However, because s is a small positive integer, it is accepted that the improvement of networks throughput and longer networks exist time at the cost of a little more packets delay.

(3) Networks Throughput in the Same Time
Since average packets transmission delay is longer, networks throughput in the same time becomes fewer accordingly.

(4) Networks Overall Throughput in Exist Time
Although networks throughput in the same time of the EFCEM is smaller, with a view to networks overall throughput in exist time, because of a longer networks exist time, overall throughput is improved.

5 Conclusion

This paper connects cooperation enforcement mechanisms in ad hoc networks with node battery, introduces credit mode, and proposes a kind of reputation mechanisms considering node battery. Then the performance of the EFCEM is compared with that of the reputation mechanisms not considering battery. From theoretical analysis and numerical simulation, the following conclusions can be derived:

(1) Comparing with the reputation mechanisms not considering battery, the EFCEM can prolong networks exist time and improve networks overall throughput.

(2) The improvement of networks throughput and longer networks exist time is at the cost of a little more packets delay, namely, average packets transmission delay of the EFCEM is a little longer than that of the reputation mechanisms not considering battery, but it is desirable when compared with the throughput gain and longer networks exist time.

In future work, more practical factors can be considered, for instance, increasing the number of nodes, or applying in other routing protocols, then analysis will be more comprehensive and have much more practicality.

References

1. Johansson, P., Larsson, T., et al.: Senario-based performance analysis of routing protocols for mobile ad hoc networks[A]. In: MOBICOM[C]. Proceedings of the Annual International Conference on Mobile Computing and Networking, Seattle, USA, pp. 195–206 (1999)
2. Marti, S., Giuli, T., Lai, K., et al.: Mitigating routing misbehavior in mobile ad hoc networks[A]. In: MOBICOM[C]. Proceedings of the Annual International Conference on Mobile Computing and Networking, Boston, USA, pp. 255–265 (2000)
3. Buttyan, L., Hubaux, J.P.: Stimulating cooperation in self-organizing mobile ad hoc networks[J]. ACM Journal for Mobile Networks and Applications (MONET) 8(5), 579–592 (2003)
4. Buttyan, L., Hubaux, J.P.: Nuglets: a virtual currency to stimulate cooperation in self-organized mobile ad-hoc networks[R]. Federal Institute of Technology, Swiss (2001)
5. Buchegger, S., Boudec, J Y L: Performance analysis of the CONFIDANT protocol: cooperation of nodes fairness in dynamic ad-hoc networks[A]. In: (MobiHOC)[C]. Proceedings of IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing, pp. 226–236. Association for Computing Machinery, Lausanne, Switzerland (2002)
6. Michiardi, P., Molva, R.: CORE: a cooperative reputation mechanism to enforce node cooperation in mobile ad hoc network[A]. In: Communications and Multimedia Security Conference (CMS)[C]. Portoroz, Slovenia, pp. 107–121 (2002)
7. Ergen, S.C., Varaiya, P.: On multi-hop routing for energy efficiency. Communications Letters, 880–881 (October 2005)
8. Zhong, S., Chen, J., Yang, Y.: Sprite: a simple, cheat-proof, credit-based system for mobile ad-hoc networks. In: IEEE INFOCOM 2003, San Francisco, CA, USA (April 2003)

Analysis of Higher Order Voronoi Diagram for Fuzzy Information Coverage

Weixin Xie[1,2], Rui Wang[1], and Wenming Cao[2]

[1] School of Electronic Engineering, Xidian University, Xi'an, 710071, China
wrarhren@163.com
[2] ATR National Key Laboratory of Defense Technology, Shenzhen
University, Guangdong, 518060, China
{wxxie, wmcao}@szu.edu.cn

Abstract. Coverage is one of the most important issues in sensor networks. For random sensor deployment, one would like to know whether the deployed sensors can cover the whole field sufficiently. In this paper, we attempt to find a set of critical points in a sensor field so as to answer the yes/no question of complete coverage only by checking whether these points are covered, which is based on a new model of fuzzy information coverage. The higher order Voronoi diagram is analyzed and a sufficient condition is provided for a sensor field which is completely fuzzy information covered by a given set of sensors. The corresponding algorithm in *O* (K^2N+NlogN) runtime is also presented.

Keywords: Sensor network, fuzzy information coverage, higher order, Voronoi diagram.

1 Introduction

Coverage is a fundamental issue in sensor networks [1], which is a measure of service quality offered by networked sensors. In many cases, sensor coverage model is simplified as the sensing disk model [2], where a point is said to be covered if it is within the sensing area of at least one sensor. However, the sensing ability of one sensor is affected by some uncertainties in real deployment, such as the uncertain environment, the inexact values of reference standards or parameters used in the model, limited resolution and so on. In some cases, we only know the intensity about the sensing ability or incomplete information about the environment. Therefore, we propose a new model for fuzzy information coverage.

One of the coverage problems is how to determine whether the deployed sensors can sufficiently cover the sensor field. For physical coverage, there are three methods. One is to check the crossings [3] so as to examine the complete k-coverage. The authors prove that a convex region is k-covered if all intersection points between any sensors or between sensors and the boundary of the convex region are at least k-covered. Huang et al propose another method [4] for checking perimeter coverage locally to determine whether a field is sufficiently k-covered. The third one is using a k-order Voronoi diagram to check complete k-coverage [5], which has a better

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 616–622, 2007.

runtime than the perimeter coverage based algorithm [4]. After creating a k-th order Voronoi diagram $V_k(S)$ for the deployed sensor set S, the authors [5] prove that the sensor field is completely k-covered if the vertices of $V_k(S)$, the intersections between $V_k(S)$, the field boundaries, and the corners of the sensor field are k-covered.

In this paper, we attempt to find a set of critical points in a sensor field such that it's sufficient to answer the yes/no question of complete fuzzy information coverage only by checking whether these points are covered. The higher order Voronoi diagram is analyzed and a set of critical points is provided. The corresponding algorithm is also presented. The rest of this paper is organized as follows. The model of fuzzy information coverage is introduced in section 2. Section 3 provides the sufficient condition and presents the corresponding algorithm. Section 4 concludes the article.

2 Fuzzy Information Coverage

Given a sensor s_i with location (x_i, y_i) and a point p with location (x, y), the information intensity $\overline{I_i}$ at the point p affected by s_i is expressed by a membership degree. Set an effective threshold d_{th} ($1 > d_{th} \geq 0$) . $\overline{I_i}$ is called effective information intensity if $\overline{I_i} > d_{th}$. The information intensity which is less than d_{th} can be negligible. Hence, the fuzzy information coverage for K collaborative sensors is defined as follows.

Definition 1 (K-sensors, D$_{th}$-intensity fuzzy information coverage). Set a coverage threshold D_{th} ($1 \geq D_{th} > d_{th}$). For geographical distributed K sensors, a point is said to be (K,D_{th})-covered if the information intensity $\overline{Q_K}$ at the point affected by K sensors is larger than or equal to the threshold D_{th}, i.e. $\overline{Q_K} \geq D_{th}$. A region is said to be (K,D_{th})-covered if all points of the region are (K,D_{th})-covered.

Consider the characteristic of target detection with uncertainty. The information intensity is defined as the following membership function.

$$\overline{I} = \begin{cases} 0, & if\ r + r_e \leq d(s,p) \\ e^{-\lambda(d(s,p)-r)^{\beta}}, & if\ r < d(s,p) < r + r_e \\ 1, & if\ r \geq d(s,p) \end{cases} \tag{1}$$

where, $d(s, p)$ is the Euclidean distance between the sensor s and the point p, r is an inner radius, r_e is the range with uncertainty, $r + r_e$ is the effective sensing range, λ, β is a positive constant, $0 < \beta \leq 1$. Thus, a fuzzy annulus for sensing ability is formed. Correspondingly, we have $d_{th} = \exp(-\lambda r_e^{\beta})$.

To obtain the final fusion intensity $\overline{Q_K}$ at a point affected by K sensors, a fusion operator $F_K : [0,1]^K \to [0,1]$ is defined as the following formula.

$$\overline{Q}_K = F_K(\overline{I}_1,\cdots,\overline{I}_K) = \min(1, \sqrt[q]{\sum_{i=1}^{K}\overline{I}_i^{\,q}}) \tag{2}$$

where q is the influence factor, and usually $\infty > q \geq 3$. K is the number of sensors. $\overline{I}_i$ is the information intensity. The fusion operator indicates that $q \to \infty$, $\overline{Q}_K = \min(1, \max_{i\in\{1,\cdots,K\}} \overline{I}_i)$ .It's observed that the fusion operator is more realistic and flexible. A smaller value of q can be taken in an optimistic fusion strategy, while a larger value of q taken in a pessimistic fusion strategy.

Based on the model of fuzzy information coverage, one property is provided. A point, which is (k,D_{th})-covered, can be also (k+1,D_{th})-covered. Thus, when checking (K,D_{th}) fuzzy information coverage for a point, one can not stop checking (1,D_{th}), (2,D_{th}) ,…, (K,D_{th}) coverage sequentially until the point is (K,D_{th})-covered. Hence an efficiency measurement for the selection of the sensor sequence is defined as follows, which is important for designing an efficient algorithm. And a necessary condition for the most efficient sequence is given in Theorem 1.

Definition 2 (efficiency of sequence). Given two selected sequences $\ell_K = (s_1, s_2, \cdots, s_K)$ and $\ell_K^{'} = (s_1^{'}, s_2^{'}, \cdots, s_K^{'})$, the sequence ℓ_K is said to be more efficient than $\ell_K^{'}$ if $\overline{Q}_{s_i} \geq \overline{Q}_{s_i^{'}}$, for all i , $i = 1, 2, \cdots, K$, where $\overline{Q}_i = F_{s_i}(\overline{I}_{s_1}, \cdots, \overline{I}_{s_i})$.

Theorem 1: the most efficient sequence $\ell_K = (s_1, s_2, \cdots, s_K)$ should satisfy

$$\overline{I}_{s_1} \geq \overline{I}_{s_2} \geq \cdots \geq \overline{I}_{s_K} \tag{3}$$

where, $\overline{I}_{s_i}$ is an effective information intensity.

Proof. When checking (K,D_{th}) fuzzy information coverage for a point, the theorem is proved by induction. For $i = 1$, we have $\overline{Q}_{s_1} = \overline{I}_{s_1}$. Obviously the sensor s_1 is selected such that the information intensity $\overline{I}_{s_1}$ at the point affected by s_1 is the strongest one among all sensors. Now assume the selected sequence for ℓ_K is the most efficient for the first i sensors. For example, the selected sequence $(s_1, s_2, \cdots, s_i)$ satisfies that $\overline{I}_{s_1} \geq \overline{I}_{s_2} \geq \cdots \geq \overline{I}_{s_i}$ and $\overline{Q}_{s_i}$ is the most efficient. According to fusion operator, we have

$$\overline{Q}_{s_{i+1}} = \min(1, \sqrt[q]{\sum_{n=1}^{i+1}\overline{I}_{s_n}^{\,q}}) = \min(1, \sqrt[q]{\sum_{n=1}^{i}\overline{I}_{s_n}^{\,q} + \overline{I}_{s_{i+1}}^{\,q}}) \tag{4}$$

Consider another sequence $\ell_K^{'}$ which has the same sequence to ℓ_K for the first i sensors, and the (i+1)th sensor different from ℓ_K . From (4), to satisfy the inequality $\overline{Q}_{s_{i+1}} \geq \overline{Q}_{s_{i+1}^{'}}$, $\overline{I}_{s_{i+1}} \geq \overline{I}_{s_{i+1}^{'}}$ holds. Hence, the selection of the (i+1)th sensor s_{i+1} should satisfy (3).

Lemma 1: when all sensors have the same membership function, the sequence $\ell_K = (s_1, s_2, \cdots, s_K)$ satisfying the following inequality (5), is one of the most efficient sequences.

$$d(s_1, p) \le d(s_2, p) \le \cdots \le d(s_K, p) \tag{5}$$

where, $d(s_i, p) = \sqrt{(x_p - x_{s_i})^2 + (y_p - y_{s_i})^2}$, p is a point in the sensor field.

3 Analysis of Complete Fuzzy Information Coverage

3.1 Problem Formation

After deploying sensors randomly in a sensor field, one would like to know whether the deployed sensors can sufficiently cover the whole field. Given a set of deployed sensors, $\mathrm{S} = \{s_1, s_2, \cdots, s_n\}$, in a two-dimensional sensor field F. $s_i \in \mathrm{S}$ with location (x_i, y_i) . For an arbitrary point $z \in \mathrm{F}$, let the information intensity $\overline{Q}_K$ at the point z be affected by the k nearest sensors. In this case, $\overline{Q}_K$ is maximal. The question of checking whether complete fuzzy information coverage is represented below.

Is every point in the sensor field is (K,D$_{th}$)-covered? In other words, Is the information intensity $\overline{Q}_K$ at every point in the field is larger than or equal to the threshold D$_{th}$? It can be transformed to the following inequality

$$\sqrt[q]{\sum_{i=1}^{K} (e^{-\lambda(d_i - r)^\beta})^q} \ge \mathrm{D_{th}} \tag{6}$$

We attempt to find the critical points in the field F in an efficient manner such that the question can be answered by only checking whether these points are (K,D$_{\mathrm{th}}$)-covered. Inspired by the property of higher order diagram, the K nearest sensors is fixed for each Voronoi region. The following theorem provides such a set of critical points

3.2 Sufficient Condition

At first, create a K-th order Voronoi diagram V$_K$(S) for the deployed sensor set S in a convex sensor field F. $\mathrm{S} = \{s_1, s_2, \cdots, s_n\}$, n is the number of sensors. For an arbitrary sensor subset $U \subset \mathrm{S} = \{s_1, s_2, ..., s_n\}$, where there exist K sensors in U, we have the Voronoi region of U.

$$cell(U) = \{p \in \mathbb{R}^2 \mid \max_{s_u \in U} \left\| p - s_u \right\|^2 \le \min_{s_v \in S \backslash U} \left\| p - s_v \right\|^2 \} \tag{7}$$

The set cell(U) is called Voronoi cell of U. it's observed that cell(U) is the set of points in $\mathbb{R}^2$ closer to all sensors in U than to any sensor in S\U. From [6], the K-order Voronoi cells of all subsets bounded by F form a subdivision of the whole field. What's more, such a subdivision is polyhedral. Thus, the problem is considered in another way that the field F is completely (K,D_{th})-covered if each subregion ($cell(U)\bigcap F$) is (K,D_{th})-covered, i.e. the information intensity of an arbitrary point in each subregion affected by K nearest sensors in U is larger than or equal to the threshold D_{th}. The objective is to find the critical points in each subregion $cell(U)\bigcap F$ in the field such that the information intensities at the critical points are minimal. Once the minimal information intensities are all larger than or equal to the threshold D_{th}, then the region $cell(U)\bigcap F$ is (K,D_{th}) covered. Finding the critical points can be simplified as an optimization problem.

$$\begin{aligned} &\min_{(x,y)\in\mathbb{R}^2} && \sqrt[q]{\sum_{i=1}^{K}(e^{-\lambda(d_i-r)^\beta})^q} \\ &\text{s.t.} && d_i=\sqrt{(x-x_i)^2+(y-y_i)^2}, \\ & && (x,y)\in \text{each subregion cell}(U)\bigcap F \end{aligned} \tag{8}$$

where, s_i with location (x_i,y_i), $cell(U)\bigcap F$ is polyhedral. The objective function adopted is equivalent to the membership function. The following theorem provides that (1) the Voronoi vertices; (2) the intersections between Voronoi diagram and the field boundaries; (3) the vertices of the field are the critical points.

Theorem 2: (Sufficient condition for complete (K,D_{th}) coverage). Firstly, create a K-th order Voronoi diagram V_K(S) for the deployed sensor set S in a convex sensor field F. The field is said to be completely (K, D_{th}) covered if (1) the vertices of V_K(S); (2) the intersections between Voronoi diagram and the field boundaries; (3) the vertices of the field are (K, D_{th})-covered.

Proof. Based on property of K-th order Voronoi diagram, it's well known that each subregion $cell(U)\bigcap F$ is a convex set and there exist K nearest sensors in U. On the other hand, the distance function d_i is a convex function over a point, $e^{-\lambda(d_i-r)^\beta}$ ($0<\beta\le 1$) is a convex function. Due to the property of convex function, the function $e^{-\lambda(d_i-r)^\beta}$ ($0<\beta\le 1$) is also a convex function. According to Minkowski's inequality, the objective function is also a convex function. Based on theory of the convex analysis, the objective function has the extremum at vertices of each convex subregion $cell(U)\bigcap F$. So we say that all vertices of each convex subregion $cell(U)\bigcap F$ are considered to be the critical points, which consist of (1) the Voronoi vertices; (2) the intersections between Voronoi diagram and the field boundaries; (3) the vertices of the field. For each vertex of V_K(S), there exist some sequences formed by the k nearest sensors. Because of the same information intensity affected by one of the sequences, it's sufficient to only checking one sequence of the k nearest sensors.

3.3 Complete (K,D_{th}) Coverage Eligibility Algorithm

Based on the theorem above, a complete (K,D_{th}) coverage eligibility algorithm is provided. The algorithm is described below. Note that the algorithm is centralized. It needs a leader node to carry out the algorithm.

Theorem 3. For N-sensor set, the eligibility algorithm runs in $O(K^2N+N\log N)$ time

```
Complete (K,Dth) coverage eligibility algorithm
{ Given a sensor set S={$s_1$,…,$s_n$}.$s_i$ with location
    ($x_i$,$y_i$) in a sensor field F. }
Create the K-th order Voronoi diagram $V_K$(S) for
the sensor set S to obtain vertices of each
subregion $cell(U)\bigcap F$ .
For each vertex obtained (the Voronoi vertices;
  (2) the intersections between $V_K$(S) and the
  field boundaries ;(3) the vertices of the field)
    if $\overline{Q}_K$ at the vertex affected by K nearest
        sensors is less than the threshold $D_{th}$
        Return No
    end if
end for
Return Yes
```

Proof. the process is similar to that in [5].It needs $O(K^2N+N\log N)$ time to construct the K-order Voronoi diagram VK(S) by the set of N sensors using the linear-time algorithm proposed in [6]. Because of the number of vertices and edges of VK(S) bounded by $O(K(N-K))$ and limited vertices of the convex field [7], the remaining steps can be done in $O(K(N-K))$ time. The desired result is obtained.

4 Conclusions

We have analyzed the higher order Voronoi diagram for fuzzy information coverage and have proved the vertices of all subregions formed by K-order Voronoi cell and convex field are the critical points. It's sufficient to only check whether the critical points are (K,D_{th})-covered to answer the yes/no question of complete (K,D_{th}) fuzzy information coverage. And the corresponding algorithm proposed can be solved in $O(K^2N+N\log N)$ time.

References

1. Cardei, M., Wu, J.: Coverage in wireless sensor networks. In: Ilyas, M., Magboub, I. (eds.) Handbook of Sensor Networks. Ch. 19, CRC Press, USA (2004)
2. Wang, B., Wang, W., Srinivasan, V., Chua, K.C.: Information coverage for wireless sensor networks. IEEE Communications Letters 9(11), 967–969 (2005)
3. Wang, X., Xing, G., Zhang, Y., Lu, C., Pless, R., Gill, C.: Integrated coverage and connectivity configuration in wireless sensor networks. In: SenSys. ACM International Conference on Embedded Networked Sensor Systems, pp. 28–39 (2003)
4. Huang, C.-F., Tseng, Y.-C.: The coverage problem in wireless sensor networks. Mobile Networks and Applications 10(4), 519–528 (2005)

5. So, A.M.-C., Ye, Y.: On solving coverage problems in a wireless sensor network using voronoi diagrams. In: WINE. The 1st Workshop on Internet and Network Economics (2005)
6. Aggarwal, A., Guibas, L.J., Saxe, J., Shor, P.W.: A Linear-Time Algorithm for Computing the Voronoi Diagram of a Convex Polygon. Discrete and Computational Geometry 4, 591–604 (1989)
7. Lee, D.-T.: On k-Nearest Neighbor Voronoi Diagrams in the Plane. IEEE Transactions on Computers C31(6), 478–487 (1982)

Anonymous Mutual Authentication Protocol for RFID Tag Without Back-End Database

Song Han, Tharam S. Dillon, and Elizabeth Chang

DEBI Institute
Curtin Business School
Curtin University of Technology
GPO Box U1987
PERTH, WA 6845 Australia
s.han@curtin.edu.au

Abstract. RFID, as an emerging technology, has very huge potential in today's social and business developments. Security and Privacy are one of the important issues in the design of practical RFID protocols. In this paper, we focus on RFID authentication protocol. RFID mutual authentication is used to ensure that only an authorized RFID reader can access to the data of RFID tag while the RFID tag is confirmed that it releases data to the authenticated RFID reader. This paper will propose an anonymous mutual authentication protocol for RFID tag and reader. RFID tag is anonymous to RFID reader so that privacy can be preserved. In addition, mutual authentication does not need to rely on a back-end database.

1 Introduction

Radio-Frequency Identification (RFID) is emerging technology in automatic identification and tracking systems [1]. RFID systems consist of Radio Frequency (RF) tags, or transponders, and RF tag readers or transceivers. The transponders themselves typically consist of integrated circuits connected to an antenna. The use of silicon-based microchips enables a wide range of functionality to be integrated into the transponder. Typical functionality ranges from large read/write memories to integrated temperature sensors to encryption and access control functionality. The transceivers query the transponders for information stored on them. This information can range from static identification numbers to user written data to sensory data. Therefore, RFID have numerous potential applications in automatic identification and tracking purposes [1, 2, 3], such as in supply chain management benefiting industries by increasing the visibility and accuracy of the shipment data. RFID can reduce overhead and errors associated with moving items through the manufacturing steps in industrial automation. RFDI can also help to infer people's current behavior and their actions as an implicit input for computer systems in hospital systems or anti-terrorism system.

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 623–632, 2007.

Different from the older bar-code technology [15], RFID tags have a number of important advantages:

1. The small size can allow them to be implanted within objects;
2. Identification by frequency allows objects to be read in large numbers without the need for a visual contact.
3. RFID identifiers are long enough so that every object has a *unique* code. Such universal uniqueness means that a product may be tracked as it moves from location to location, finally ending up in the consumer's hands. This may help companies combat theft or improve management of stock and inventories in shops or warehouses [14].
4. The introduction of RFID tags in all objects could also directly benefit the consumer: waiting times at checkout lines may be drastically reduced by the use of reader technology hat requires no bar-code scanning.

Thanks to these advantages of RFID technique, RFID tags have been used in transport systems, passports, automotive, animal tracking, Human implants, RFID in library, and so on. The following will introduce the components of an RFID system and the related security and privacy issues.

1.1 RFID System

General back-end database based RFID systems are comprised of four components:

(1) the RFID tag, or transponder, which is located on the object to be identified and is the data carrier in the RFID system;
(2) the RFID reader, or transceiver, which may be able to both read data from and write data to a transponder;
(3) the back-end database server, which helps with registrations and authentication of RFID tags; and
(4) the data processing subsystem which utilizes the data obtained from the transceiver in some useful manner.

1.2 Security and Privacy of RFID

The security infrastructure surrounding this technology, balancing privacy against commercial applications, is a major concern. RFID is operated on radio frequency for reading tags and does not require line-of-sight operations. Consider a supermarket scenario: a supermarket can wave a RFID reader near people's clothes, handbags, and other personal items and retrieve private data about people and their belongings and shopping behavior. The size of their shirts is no longer their personal secret, nor is the amount of cash they are carrying. Therefore, it is possible to create a complete commercial, and, worse, personal profile with the collected tag data on the person. One may argue that this problem would be solved by disabling each tag when it leaves a shop, but this is neither in the interest of the retailer nor, in fact, of the customer. Tags will store warranty information (paper receipts will no longer be accepted) and, more important, the rights of ownership (a stolen item is easily identified). It is thus in the interest of the purchaser to keep tags alive. Furthermore, using RFID-enabled phones in the exchange of goods and money is both a guarantee and a proof of the transfer of ownership. Only a troublemaker would want to destroy or disable such useful data.

In this paper we will focus on the privacy issue and RFID tag authentication. In fact, the mass deployment and acceptance of RFID technology is nowadays mainly limited by privacy concerns [1, 4, 6, 7]. Products labeled with RFID tags reveal sensitive information when queried by readers, and they do it indiscriminately. This may induce the violation of location privacy, i.e. tracking. Besides the privacy and tracking issues, there are some other worth mentioning: impersonating, spoofing, eavesdropping, traffic analysis, etc..

1.3 Advantages of the Proposed Protocol

In this paper, we will propose a secure authentication protocol for RFID tag and reader without back-end database. The advantages of the new protocol are as follows:

- The authentication process does not involve a back-end database server with consistent connection with the RFID reader. Therefore, the authentication is a serverless authentication.
- The authentication is mutual between RFID reader and tag, i.e. the RFID reader is authenticated to the RFID tag while the latter is also authenticated to the former.
- There is an off-line registration authority that is responsible for preparing the initial stages for RFID tag and reader.
- The unique identity ID of RFID tag is encapsulated from authorized RFID readers. Therefore, the privacy of the RFID tag and its owner is preserved. This point is held from two aspects: the identity privacy is preserved from not only the authorized RFID readers but also any adversary from outside.
- The unique identity of the RFID tag can be revealed in case of dispute. This is supported by the fact that the off-line registration authority and the RFID tag share a pre-established secret. Therefore, the off-line registration authority will release the unique identity of the compromised or misfunctioned RFID tag to the RFID reader or a legal third party.

1.4 Organization of the Rest of the Paper

The rest of the paper is organized as follows: In Section 2, some related works based on the back-end database model will be introduced. A previous work that was serverless will also be briefly reviewed. In Section 3, a secure anonymous mutual authentication protocol for RFID tag without back-end database will be proposed. This section is composed of three subsections: the first is notation introduction; the second is the initial stage for the RFID tag, the RFID reader and the off-line registration authority; the third will be the proposed mutual authentication protocol. In Section 4, the security analysis and comparison will be provided. Finally, we will conclude our paper.

2 Related Works

In this section, we will review two kinds of existing RFID authentications. One is the back-end database server model based authentication [5, 6, 9, 13], the other is the non-server authentication [12]. The back-end database server based authentication resorts to a back-end database server to help an authorized RFID reader to authenticate RFID tag and vice versa. The non-server authentication does not resort to any online help besides the RFID reader and the RFID rag. On the other hand, the non-server authentication protocol only needs an off-line registration authority. Therefore, a consistent connection between the RFID reader and a trusted third party (say, the registration authority in the non-server model) is removed.

The proposed protocol in [10] provides specific time-memory trade-off that supports the scalability. The authors also proved that the system could truly offer privacy and even forward privacy. The authors further provided an extension of the scheme which offers a secure communication channel between RFID tags and their owner using building blocks that are already available on the tag.

The protocol proposed in [11] aims to solve the desynchronization problem by maintaining a previous ID in the database server. However, an adversary who queries T actively without updating ID can trace the RFID tag because the hashed ID is continually identical. Moreover, the adversary can trace the previous event of tag because ID is updated b XORing the previous ID with a random number emitted through radio frequency.

The authentication protocol in [4] was based on hash-chain. It only requires a hash function in the tag and data management at the back-end. It offers a high degree of location privacy and is resistant to many forms of attacks. Further, only a single message exchange is required, the communications channel needs not be reliable and the reader/third party need not be trusted, and no long-term secrets need to be stored in tags. However, their solution did not provide full privacy guarantees; i.e. the tag is vulnerable to tracing when the attacker interrupts the authentication protocol mid-way.

A hash-tree based authentication protocol for RFID tags was proposed in [5]. The authors gave a general scheme for building private authentication with work logarithmic in the number of RFDI tags based on a scheme with linear work as a sub-protocol. However, the amount of computation required per tag is not constant, but logarithmic with the number of tags in the hash-tree.

An anonymous RFID protocol was proposed in [9]. This protocol enforces privacy as it prevents information to be read by unauthorized third parties. This is due to the fact that no single, fixed ID is used throughout the tag's life as tag IDs get refreshed periodically. This protocol also offers a high degree of location privacy and is resistant to many forms of attacks.

Han et al. [8] proposed a new mutual authentication protocol for RFID tags. The RFID reader and tag carry out the authentication based on their synchronized secret information. The synchronized secret information is monitored by a component of the database server. Their protocol also supports the low-cost non-volatile memory of RFID tags. This is desirable since non-volatile memory is an expensive unit in RFID tags. However, their protocol still needs the back-end database support.

All the above authentication protocols were based on back-end database server. Therefore, a consistent connection between the RFID reader and the back-end database server needs to be maintained in those protocols. In order to remove the requirement of such consistent and secure connection, Tan et al. proposed a non-server authentication protocol [12]. However, their protocol did not support mutual authentication between RFID tag and reader. In addition, the anonymity of RFID tags was not maintained while anonymity is one of the important properties concerned in ubiquitous computing environment [2].

In the next section, we will propose a new authentication protocol for RFID tag and reader. This protocol does not need to maintain a back-end database server, and thus can remove the secure and consistent connection between the RFID reader and its back-end database server. The proposed protocol supports mutual authentication for RFID reader and tag. It also maintains the privacy of the RFID tag and its owner. The privacy can be disclosed by an off-line trusted third party, i.e. the registration authority of the RFID system.

3 New Anonymous Mutual Authentication Protocol for RFID Tags

In this section, we will present the anonymous mutual authentication protocol for RFID tags. Some notations will be first presented and then used throughout the rest of the paper. The structure of the RFID reader and the off-line registration authority will be then introduced. Following that, the anonymous mutual authentication protocol for RFID tags will be provided.

3.1 Notations Used in Our Protocol

The following table provides the notations used in the proposed mutual authentication protocol.

Table 1. Notations for the serverless authentication protocol for RFID tag and reader

Notation	Representation
h()	One-way hash function available to all parties
||	Concatenation of bit-strings
T	A valid RFID tag
CA	Off-line registration authority
R	An authorized RFID reader
id_T	Unique identity of T
PRNG	A pseudorandom number generator
β	The bit-length of the output of h()
$\oplus$	Exclusive-or function (XOR)
id_R	Unique identifier of R
s	A secret of T which is shared with CA
$h(id_T)$	Pseudo-identifier of T
L	Authentication list of R

3.2 Initial Preparations

The RFID tag T has a one-way hash function h(). T can calculate the hash value for any input to this function. T can also carry out the XOR calculation. In fact, carrying out the XOR calculation is an affordable capability for various RFID tags, especially for low-cost RFID tags. The tag T also shares a secret key with an off-line registration authority CA. This secret key is assigned to T while it is registered with the CA. The RFID reader needs to register at the registration authority which will assign a list of hash valuation of identities of RFID tags to the RFID reader. That is to say, the reader R is authorized to have rights to access the data of those RFID tags.

The structure of the authentication list of the RFID in the j-th authentication process for the i-th tag T

…	…	…
$h(s,id_R)$	h(id$_T$)	j
…	…	…
…	…	…

Fig. 1. The structure of the authentication list. We use one RFID tag instance T to demonstrate the components of the authentication list of RFID reader. id$_T$ is the unique identifier of T. The secret key of T is s which is assigned by the off-line registration authority. id$_R$ is the unique identifier of the RFID reader.

3.3 The Proposed Protocol

This authentication protocol is working without the timestamps. The details of the mutual authentication protocol are presented as:

1. RFID reader R sends an access Request to RFID tag T.
2. T checks the request and generates a nonce r_1. T then sends r_1 back to R.
3. After receiving r_1, R generates a new nonce r_2 and sends r_2 and its identifier id_R to T.
4. T uses hash function h(), its secret s, and R's identifier id_R to get $h(h(s, id_R))$, chooses the first m bits of it to get t_1, and then sends t_1 to R.
5. R searches his list L for finding a $h(x, id_R)$ such that the first m bits of $h(h(x, id_R))$ is identical to t_1. R then computes $f_1 = h(h(s,id_R) \| t_1 \| r_2)$ and sends it to T.
6. After receiving t_1, T sets $f_2 = h(h(s,id_R) \| t_1 \| r_2)$ and compares the received f_1 with f_2. If they are equal, then R is authenticated and T believes R is authorized to access to T. Following that, T first encapsulates its unique identity id_T to get a pseudo-identifier and then encodes the pseudo-identifier to get $f_3 = h(h(s,id_R) \| r_1 \| r_2) \oplus h(id_T)$. T finally forwards f_3 to R.

7. After receiving f_3, R computes $f_4 = h(h(s,id_R) \parallel r_1 \parallel r_2))$ and then sets $f_5 = f_3 \oplus f_4$. If the tag is a valid tag which R is authorized to access, then this f_5 is the encapsulated identity of T. To confirm this point, R checks his list L. If f_5 is in L, then T is authenticated and R believes T is a valid RFID tag.

The serverless mutual authentication protocol for RFID tag with encapsulated ID is summarized in the following figure.

Reader (PRNG, h(), L)		Tag (PRNG, h(), s)
	Request →	
		generates r_1= PRNG
	← r_1	
generates r_2 = PRNG		
	id$_R$, r_2 →	sets $t_1 = [h(h(s,id_R))]_m$
	← t_1	
checks L for matching t_1 computes $f_1 = h(h(s,id_R) \parallel t_1 \parallel r_2)$		
	f_1 →	sets $f_2 = h(h(s,id_R) \parallel t_1 \parallel r_2)$ checks whether $f_1 = f_2$ If yes, then R is an authorized reader, and sets $f_3 = h(h(s,id_R) \parallel r_1 \parallel r_2) \oplus h(id_T)$
	← f_3	
computes $f_4 = h(h(s,id_R) \parallel r_1 \parallel r_2))$ sets $f_5 = f_3 \oplus f_4$ searches L for confirming f$_5$ is in L then T is a valid tag		

Fig. 2. Mutual authentication protocol for RFID tag with encapsulated ID. In the process of authentication, there is no third party involved. Also, the authentication protocol is mutual. After the successful mutual authentication, the RFID reader R gets the pseudo-identifier $h(id_T)$.

Remark 1. The authentication list L maintained by the RFID reader is assumed to be secure against both passive and active attacks. Otherwise, an adversary who may modify the authentication list so that any legal RFID tag cannot correctly authenticate the RFID reader while any illegal RFID tag may be correctly authenticated by the RFID reader.

3.4 Revocation of Anonymity of RFID Tags

The identity id_T of the RFID tag is encapsulated with a one-way hash function. Therefore, anonymity is maintained with respect to both the authorized RFID reader and any adversary outside. However, the anonymity can be eradicated by the registration authority. This property is useful in case of dispute. Consider such a scenario where a valid RFID tag which has registered with the off-line registration authority, is compromised by an adversary or the tag is misused by a malicious owner, then it is necessary to remove the anonymity and identify the real identity of the RFID tag. The registration authority can do this because the registration authority shares a secret s with a valid and registered RFID tag. In summary, the registration authority does not involve the online authentication between the RFID reader and tag. It only participates the initial stage and the revocation of anonymity of RFID tags.

4 Security Analysis

This section will provide the security analysis and a comparison of security characteristics between the new protocol and a previous protocol.

4.1 Security Analysis

Mutual authentication: The authentication between RFID tag T and reader R is mutual authentication. T is authenticated to R by matching f_5 with an element in the authentication list L. R is authenticated to T with checking whether $f_1 = f_2$.

Attack on the RFID tag: If an adversary tries to impersonate an authorized RFID reader and attack on the RFID tag, then the adversary does not know the secret information s. As a result, the adversary cannot compute f_1. Therefore, the adversary cannot let the RFID tag accept its access.

Attack on user privacy: From the construction of our protocol, we can see that no identity was released by an RFID tag participating in the authentication process. Therefore, eavesdropping is not an issue as long as the hash function h() is one-way hash function so that no usable information on its pre-image is revealed. As a result, user privacy is guaranteed.

Anti-cloning: If an adversary tries to make cloning by replaying intercepted interactions, then the adversary cannot success cloning because r_2 is different for each authentication run. On the other hand, the adversary cannot work out a valid $f_3 = h(h(s, id_R) \| r_1 \| r_2) \oplus h(id_T)$ without knowing the secret key s as well as the identity of the RFID tag.

4.2 Comparison

The authentication in our paper is a mutual authentication protocol without back-end database. The protocol in [12] is also of serverless authentication. However, our new protocol is different from that authentication protocol in the following aspects:

- The new authentication protocol in our paper is of mutual authentication. That is, the RFID tag is authenticated by the RFID reader, while the latter is also authenticated by the former. The protocol 2 of [12] was not of mutual authentication. In fact, that protocol 2 only provided the authentication from RFID tag to reader. However, the authentication from RFID reader to tag is also important. This is because in some scenarios it can help to confirm the RFID tag believes it is communicating with an authorized RFID reader.
- In our authentication protocols, the RFID tag only transmits its encapsulated identity to an authorized RFID reader. The authorized RFID reader only gets the pseudo identifier of the tag. This property keeps the anonymity of the RFID tag's identity from not only the authorized RFID reader but also any adversary outside.
- The anonymity of RFID tags can be revealed by the off-line registration authority upon request. This is supported by the fact that the RFID tag has registered with the off-line registration authority.

5 Conclusions

Mutual authentication architecture without back-end database enables the removal of reliable consistent connections between RFID readers with their database server. A mutual authentication protocol without back-end database for RFID readers and tags has been proposed in this paper. The protocol enables not only RFID tag to authenticate RFID reader but also the latter to authenticate RFID tag. The second property of the proposed protocol is the identity of RFID tag has been encapsulated so that the anonymity of RFID tags (and their owners) is preserved. It is useful in a ubiquitous computing environment where users may concern their privacy.

Acknowledgement

The authors thank the anonymous reviewers for their comments. This work is partially supported by the Curtin IRG Grant and ARC DP Grant.

References

1. Rieback, M.R., Crispo, B., Tanenbaum, A.S.: The evolution of RFID security. IEEE Pervasive Computing, 62–69 (January - March 2006)
2. Greenfield, A.: Everyware: The Dawning Age of Ubiquitous Computing. Peachpit Press, Berkeley, CA (2006)

3. Kaps, J.P., Gaubatz, G., Sunar, B.: Cryptography on a speck of dust. IEEE Computer, 38–44 (February 2007)
4. Henrici, D., Muller, P.: Hash-based Enhancement of Location Privacy for Radio-Frequency Identification Devices using Varying Identifiers. In: Proceedings of Workshop on Pervasive Computing and Communications Security (2004)
5. Molnar, D., Wagner, D.: Privacy and Security in Library RFID Issues, Practices, and Architectures. ACM Conference on Computer and Communication Security (2004)
6. Sarma, S.E., Weis, S.A., Engels, D.W.: RFID Systems and Security and Privacy Implications. In: Kaliski Jr., B.S., Koç, Ç.K., Paar, C. (eds.) CHES 2002. LNCS, vol. 2523, pp. 454–470. Springer, Heidelberg (2003)
7. Sean, W., Thomas, L.: Automatic identification and data collection technologies in the transportation industry: Barcode and RFID. Technical Report (2001)
8. Han, S., Potdar, V., Chang, E.: Mutual authentication protocol for RFID tags based on synchronized secret information with monitor. In: Gervasi, O., Gavrilova, M.L. (eds.) ICCSA 2007. LNCS, vol. 4707, pp. 227–238. Springer, Heidelberg (2007)
9. Dimitriou, T.: A Lightweight RFID Protocol to Protect against Traceability and Cloning Attacks. International Conference on Security and Privacy for Emerging Areas in Communication Networks- SecComm (September 2005)
10. Avoine, G., Oechslin, P.: A Scalable and Provably Secure Hash Based RFID Protocol. In: The 2nd IEEE International Workshop on Pervasive Computing and Communication Security, Kauai Island, Hawaii, USA, pp. 110–114 (March 8, 2005)
11. Lee, S.M., Hwang, Y.J., Lee, D.H., Lim, J.I.: Efficient Authentication for Low-Cost RFID Systems. In: International Conference on Computational Science and Its Applications - ICCSA, pp. 619–627 (May 2005)
12. Tan, C.C., Sheng, B., Li, Q.: Serverless Search and Authentication protocols for RFID. In: Proceedings of Pervasive Computing Conference (2006)
13. Peris-Lopez, P., Castro, J.C.H., Estevez-Tapiador, J.M., Ribagorda, A.: EMAP: An Efficient Mutual-Authentication Protocol for Low-Cost RFID Tags. In: OTM Workshops (1), pp. 352–361 (2006)
14. Wal-Mart Details RFID Requirement (November 6, 2003), Article appears in http://www.rfidjournal.com/article/ articleview/642/1/1/
15. Advanced Barcode Technology, http://bar-codes.com/

ISMS-MANET: An Identifiers Separating and Mapping Scheme Based Internet Access Solution for Mobile Ad-Hoc Networks[*]

Ping Dong, Hongke Zhang, Deyun Gao, and Xiaohua Chen

Beijing Jiaotong University, 100044 Beijing, China
{05111042, hkzhang, gaody, 05111040}@bjtu.edu.cn

Abstract. Providing easy access to the Internet is a prerequisite for successful deployment of Mobile Ad-hoc Networks (MANET). This paper proposes an Identifiers Separating and Mapping Scheme (ISMS) based Internet access solution for MANET, called ISMS-MANET. Firstly, ISMS, which is a candidate for the future Internet architecture, is briefly described. Then, how the Ad hoc On-Demand Distance Vector (AODV) routing protocol can be used for connecting a MANET to the ISMS based Internet is discussed. In ISMS-MANET, the Access Router (AR) implements AODV protocol and works like a normal MANET node. The node in the MANET relies on the Route Reply message of AODV to find the AR. A prototype system is implemented. Performance analysis shows that ISMS-MANET has much faster handover and lower packet processing delay than the Mobile-IP based solution.

1 Introduction

Mobile ad-hoc networks (MANET) are wireless networks that have dynamic, sometimes rapidly changing, random, multihop topologies. MANETs are established and maintained by the mobile nodes without relying on any fixed routers. The mobile nodes agree upon forming a spontaneous, temporary network in which they relay packets for other nodes that wish to communicate but are out of the wireless range.

With the extensive use of palm-top computers and cellular phones, it is apparent that there are more and more wireless terminals and networks exist which will facilitate the development of MANETs. For a widespread and successful deployment of MANETs, the ability to provide easy access to the Internet is a prerequisite. It will enable a user to remotely access the Internet anywhere and anytime. Connecting to the Internet through the nodes of a MANET is a very challenging and necessary issue. However, most works about the MANET has

[*] This work is supported by the National Basic Research Program of China (No.2007CB307101, No.2007CB307106), National Natural Science Foundation of China (No.60473001), and National Natural Science Foundation of Beijing City (No.4062026).

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 633–644, 2007.

been focused on isolated MANETs, not much work has been done concerning the integration of the MANET and the Internet.

To solve the Internet access issue for MANET, there are several issues that have to be considered, such as location management, handover, and routing. The nodes in MANETs can communicate with each other by using specific routing protocols, such as Ad hoc On-Demand Distance Vector (AODV) [1], Optimized Link State Routing Protocol (OLSR) [2], and Dynamic Source Routing Protocol (DSR) [3]. However, when a mobile node in a MANET communicates with a fix node in Internet, it is necessary to find the gateway to access the Internet. For the fix node, it is necessary to locate the mobile nodes and follow it in its motion, which introduces complications in routing protocols.

In this paper, we propose an Internet access solution for MANETs, called ISMS-MANET. Instead of Mobile IPv4/IPv6 or Network Address Translation (NAT), an Identifiers Separating and Mapping Scheme is adopted to achieve the mobility management in the Internet. Then, we focus on the issue of how MANETs, in which on-demand routing is used, can be connected to the Internet. Our solutions has been analyzed by means of an implementation of prototype system, and the results show that it can provide a faster handover and lower processing delay than previous works.

2 Related Works

There are a few existing solutions to provide Internet connectivity to the MANET. These solutions include integrating Mobile IPv4/IPv6 with MANET specific routing protocols or implementing NAT on each Internet gateway node in the MANET.

Johnsson et al. [4] presented a solution for connecting a MANET, in which on-demand routing (AODV) is used, to the Internet by using Mobile IP. Their system is called MIPMANET where MIP foreign agent care-of addresses and reverse tunneling is used to support mobile nodes in a MANET. However, the main result of MIPMANET is the effect of using unicast or broadcast transmissions for periodic agent advertisements. A similar solution to MIPMANET is suggested by Nilsson et al. [5].

M. Benzai et al. [6] proposed a hierarchical architecture (i) connecting Ad-hoc networks to the Internet and (ii) integrating Mobile IP and OLSR to manage universal mobility. Their experiments evaluate the impact of mobility on the available throughput, but the experimental results are not compared with any other similar solutions that integrate MANET with Internet.

Existing solutions for MANET Internet access are mainly based on modifying Mobile IPv4 [7] or Mobile IPv6 [8]. However, although the Mobile-IP standard presents a basic solution for mobility support in the Internet, it is not suitable for Internet access in a multihop MANET. Firstly, whenever a mobile node moves from one access point to another which belongs to a new foreign subnetwork, it has to change to a new care-of address and register the change of its access point through the new foreign subnetwork to its home network. Each handoff between subnetworks would cause a huge amount of control overhead both in MANET and in Internet. The registration process is very long compared to the link-layer handoff time, especially if the mobile node is far from its home network. Secondly, Mobile IP was designed to have the foreign agent and the mobile node on the same link. In a MANET, the foreign agent and a mobile node might not have link-layer connectivity but have to use multihop communication. In this scenario a mobile node cannot determine if a foreign agent is reachable by using link-layer feedback anymore. Thirdly, traditional IP Duplicate Address Detection (DAD) used by Mobile IP is not applicable for MANET either because of the multihop problem or DAD time bound [9].

NAT-based Internet connectivity for on-demand MANETs was studied in [10, 11], in which the NAT-module of the gateway translates the packets sent by the nodes in MANET before they are forwarded onto the Internet. However, with a NAT-based gateway, all outgoing packets belonging to the same communication session must pass through the same gateway, otherwise the session will break. Besides, completely unrestricted end-to-end communication can not be achieved under the NAT-based solution.

3 ISMS: Identifiers Separating and Mapping Scheme

When the current Internet architecture was originally designed in the late 1970's and early 1980's, neither mobility nor security were considered because it was hardly imaginable that most of the world's computers would eventually be mobile and work in an distrustful environment at the same time. Thus, the protocol suite was designed with statically located hosts and without any security precautions. However, the assumptions of the architecture are being challenged by new problems within the last ten years. It may be necessary to do some radical reengineering for the architecture to bring the Internet architecture in par with the new requirements, because the fundamental shortcomings which can hardly be solved by just adding new functionality.

Many of the existing attempts at improving the Internet architecture, such as Mobile IP and NAT, are trying to ”patch” it through a series of techniques which are irrelative with each other. The result has not been a flexible and extensible platform for future internetworking. The approach presented in this section is different from that multiple technologies. The goal of our approach [12] is to build an architecture that can address mobility, multi-homing and related security at the same time, to provide end-to-end communication in heterogeneous internetworking environments, and to treat dynamic changes in a scalable manner and increase the level of protection against privacy disclosures. Our architecture employs identifiers separating and mapping scheme, access routers, and specific protocol constructs to address these issues.

3.1 Assumptions

This section describes the main assumptions that underlie the Identifiers Separating and Mapping Scheme (ISMS).

The first assumption is that each terminal has at least one unchanged identity which is named as Accessing Identifier (AID) and serves as the identity of the terminal. ISMS uses AIDs similarly with how Host Identity Protocol (HIP) [13] uses host identities: decoupling node identities from their network locations and providing a firm foundation for security. But, different from HIP, the terminals in this architecture do not know their own locators named as Routing Identifier (RID) which are managed by the network. There is a separating and mapping scheme between the terminal's AIDs and RIDs which is performed by the access router and identifiers mapping server which are located within the network.

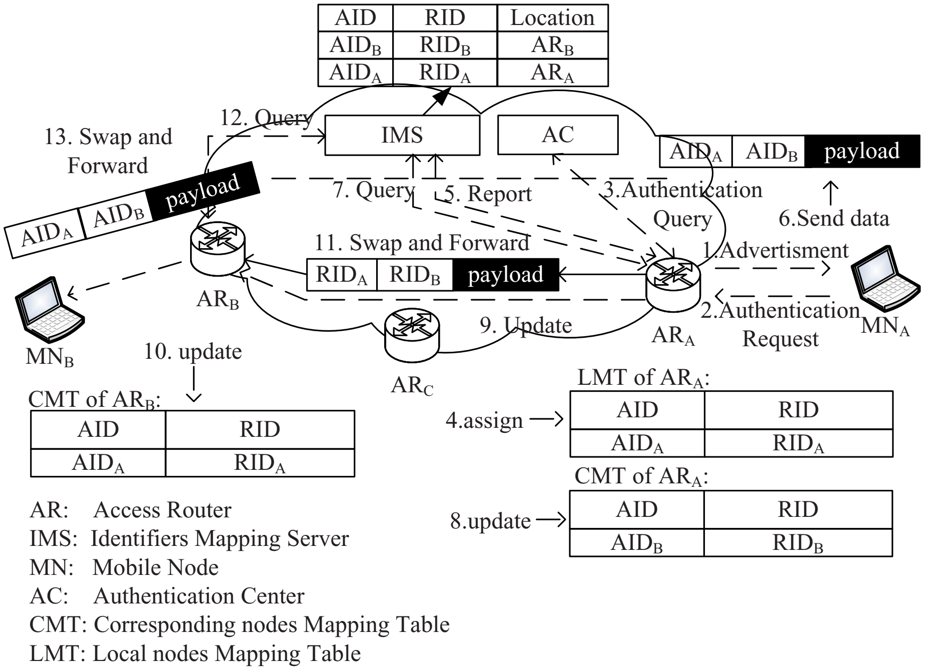
A second assumption is that connectivity between different domains is dynamic. It means that each individual domain could be a mobile network. A related assumption is that the nodes which belong to one or more domains may similarly move from some domains to others, while concurrently remaining part of other domains. Generally, for connectivity between domains and between nodes, these events happen independently from each other.

These assumptions are the foundations of the following sections where end-to-end connectivity between nodes across a dynamically domains and nodes is discussed.

3.2 Identifiers Separating and Mapping Scheme

This section presents the Identifiers Separating and Mapping Scheme, starting with an example and showing how routing and identifiers resolution occur. Fig.1 depicts the basic steps of our proposed identifier interchange procedure when MN_A sends packets to MN_B.

The packets in Fig.1 show only the source and destination address field and the payload. The detailed sequence of the procedure is numbered in the figure. (1)When an MN (MN_A) enters the area of AR_A, it receives an "Advertisement" from the AR_A. (2) MN_A sends "Authentication Request" message to AR_A. (3) AR_A sends "Authentication Query" to Authentication Center (AC) for MN_A. (4) If MN_A is valid, AR_A assigns a RID (RID_A) for the MN_A, stores the $<AID_A, RID_A>$ pair in its Local nodes Mapping Table (LMT), (5) and informs the Identifiers Mapping Server (IMS) of the $<AID_A, RID_A>$ pair. The IMS will store this pair and the location of MN_A in its database. (6) When MN_A sends packets to MN_B, it first sends the packets to the AR_A with the AID_B of MN_B as the destination address and AID_B of MN_A as the source address. (7) When the packets reach AR_A, AR_A queries the IMS for the <AID, RID> pair of MN_B. (8) When IMS responds, AR_A stores the $<AID_B, RID_B>$ pair in its Corresponding nodes Mapping Table (CMT). (9) AR_A sends "Update", which includes the information of the $<AID_A, RID_A>$ pair, to AR_B. (10) When AR_B receives "Update" from AR_A, AR_B stores the $<AID_A, RID_A>$ pair in its CMT. (11) AR_A refers to LMT/CMT and swaps the source and destination address of the packet from the AID to the RID. In this way, the packet is routed to AR_B by using the RID with no encapsulation on the optimized route. (12) If AR_B



AID	RID	Location
AID_B	RID_B	AR_B
AID_A	RID_A	AR_A

CMT of AR_B:

AID	RID
AID_A	RID_A

LMT of AR_A:

AID	RID
AID_A	RID_A

CMT of AR_A:

AID	RID
AID_B	RID_B

AR: Access Router
IMS: Identifiers Mapping Server
MN: Mobile Node
AC: Authentication Center
CMT: Corresponding nodes Mapping Table
LMT: Local nodes Mapping Table

Fig. 1. Basic steps for packet routing procedures

has no knowledge of the <AID, RID> pair of MN_B (because step 9 and step 10 might not be finished), it will query the IMS for the <AID, RID> pair of MN_A. (13) Then the RID_A and RID_B are swapped to AID_A and AID_B respectively. Finally, the packet is sent to MN_B with the AIDs of MN_A and MN_B by AR_B.

As described above, in ISMS, at the start of packets routing, the originating AR queries IMS for the <AID, RID> pair of corresponding node so that it can send packets directly to the corresponding AR. Thus, packets are always transferred on the optimized route constantly. Since the hosts must communicate with each other by using AIDs, the nodes inside of the network (routers and management nodes), which communicate with each other using RIDs, are protected from the accessing and attacking of the hosts. Since the RIDs which hold the mobile hosts' location information is hidden in the network, the location privacy of the host is protected intrinsically. In addition, since the AR works together with the AC and IMS to authenticate and management the mobile hosts by using AID as the key, the RID of AC/IMS never needs to notify to the mobile hosts. Further more, since the AID which serves as the identity of a mobile host never changes whichever network the mobile host connects to, the problems brought by IP DAD [9] will never exist. Finally, since identifier swapping is used instead of encapsulation, packets are transferred efficiently in the wireless section as well as in the wired section.

4 ISMS-MANET

ISMS-MANET is designed to provide nodes in MANET with access to the Internet. In this solution, ISMS ARs serve as the access points for MANET nodes to connect to the Internet. The MANET routing protocol is used to deliver packets between the ARs and the mobile nodes.

4.1 Proposed Architecture for ISMS-MANET

The proposed architecture is depicted in Fig.2. The MANET is interconnected to Internet via one or multiple ARs. The motion of mobile nodes inside an MANET is managed by the AODV protocol. Mobility between different MANETs or subnetworks is managed by ISMS.

The architecture is composed of several functional entities:

o Access Router (AR): A router which serves as the Internet gateway for the mobile nodes in MANET. It assigns and maintains the <AID, RID> pairs for local mobile nodes, swaps the source and destination address of packets from AID(RID) to RID(AID), and forwards the packets to its destination. AR also implements the AODV protocol to contribute to mobility and multi-homing management inside MANETs.

o Mobile Nods (MN): A mobile node in the MANET having one or more wireless interfaces. It implements the AODV and uses its AID to establish and maintain routes inside the MANET. It finds the ARs by RREP message and maintains an AR list for the ARs which it has registered with. It can communicate with nodes outside of the MANET through any of the ARs in its AR list.

o Identifiers Mapping Server (IMS): Just as DNS or RVS [13], there are multiple IMS in the whole Internet. One IMS maintains all the <AID, RID> pairs of fixed nodes and mobile nodes in a certain area. It contributes to the macro-mobility management and location management of mobile nodes.

o Authentication Center (AC): It works like the AUC in 3G network and manages the authentication information of all the nodes in a certain area. In order to concentrate our discussion on the main purpose of this article, we will not discuss the function of AC in detail.

The operation of this architecture is as follows.

1) Nodes in an MANET use their AIDs for all the communication and register with one or more AR(s). Wherever a node moves, its AID remains unchanged.

2) To send a packet to a host on the Internet: Send the packet following a route through one of the ARs with whom the mobile node has been registered. By the use of ISMS, soft handover and multi-homing approach is supported.

3) To receive packets from hosts on the Internet: The packets are routed to the AR by ordinary ISMS mechanisms discussed in section 3. The AR will then deliver the packets to the node in the MANET.

4) Nodes that do not require Internet access will see the MANET as a standalone network, i.e., they will not need any knowledge about routes to destinations outside of the MANET.

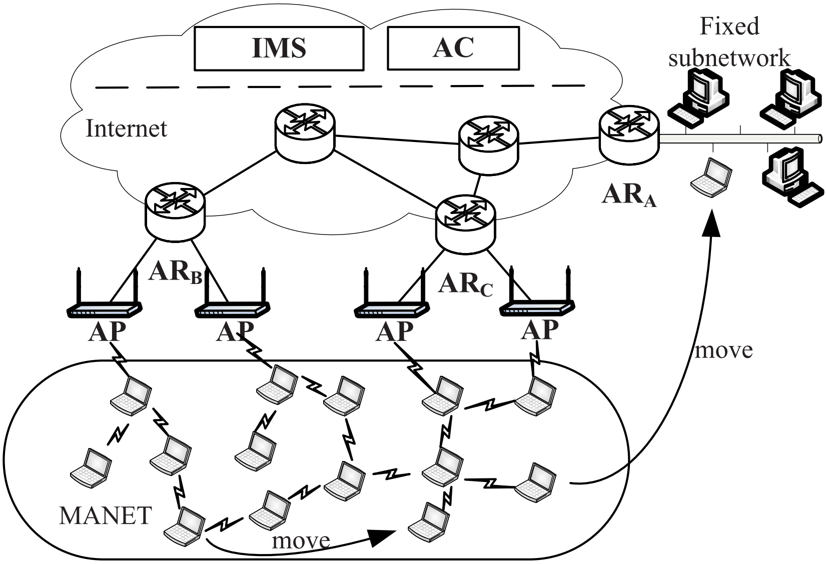


Fig. 2. MANET interconnected to an ISMS based Internet

4.2 Integration of ISMS and AODV

The proposition described in this section allows the MANET node to follow the same procedure to visit a node in Internet as a node in MANET.

When a node in a MANET wants to communicate with another node, it is impossible to decide whether the destination is located within the same MANET or not by simply looking at the destination address. To solve this problem, Mobile IP or NAT based MANET Internet access solution lets the route discovery mechanism of the MANET routing protocol search for the destination within the whole MANET network before it can be decided whether the destination should be visited through the Internet gateway or not. If the destination is not in the MANET, the searching procedure will last for a long time and route discovery messages will waste available network resources which are limited in MANETs.

In ISMS-MANET, this problem does not exist inherently. By searching the LMT/CMT maintained by AR and querying the IMS, an AR can easily obtain the information that whether a destination is reachable or not in Internet. When an AR receives the RREQ from an originate node in MANET, if the destination can be reached though the AR, the AR will work just like a normal node in the MANET and sends RREP to the originate node. After receiving the RREP message, the originate node adds a host route in its route table for the destination. Then, it registers with the AR and sets up communication with the destination. In this procedure, a originate node in a MANET does not have to search within the whole MANET before it can determine weather the destination is in the MANET or not. Fig.3 shows the processing procedure when an AR receives RREQ message.

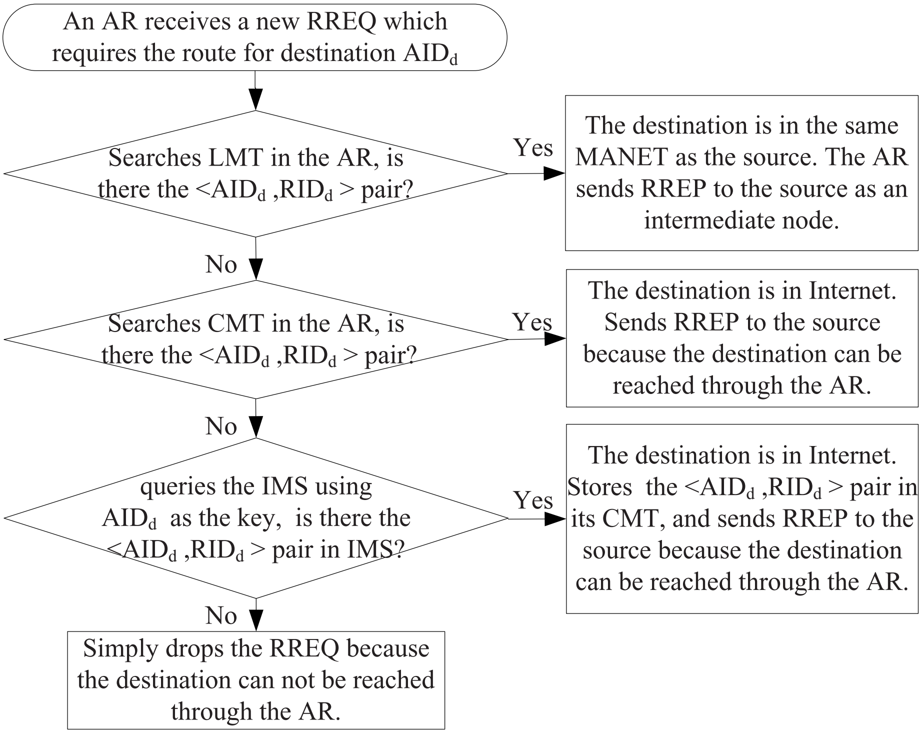

An AR receives a new RREQ which requires the route for destination AID$_d$
Searches LMT in the AR, is there the <AID$_d$,RID$_d$ > pair?
Yes
The destination is in the same MANET as the source. The AR sends RREP to the source as an intermediate node.
No
Searches CMT in the AR, is there the <AID$_d$,RID$_d$ > pair?
Yes
The destination is in Internet. Sends RREP to the source because the destination can be reached through the AR.
No
queries the IMS using AID$_d$ as the key, is there the <AID$_d$,RID$_d$ > pair in IMS?
Yes
The destination is in Internet. Stores the <AID$_d$,RID$_d$ > pair in its CMT, and sends RREP to the source because the destination can be reached through the AR.
No
Simply drops the RREQ because the destination can not be reached through the AR.

Fig. 3. AR's RREQ process procedure in ISMS-MANET

Before a MANET node can communicate with its corresponding node through an AR, it must register with the AR to pass the authentication and be assigned at least one <AID, RID> pair. We use a new flag called the 'I' flag in the RREP to indicate that the MANET node has not been authenticated to connect to the Internet. After receiving the RREP with the 'I' flag set to zero, the originate node sends authentication request to the AR to finish the register procedure. After these procedures, the source node can communicate with the nodes outside of the MANET which it attaches to.

5 Tests and Measurement

To evaluate how the ISMS designs and their characteristics will influence protocol performance such as handover and confirm that the ISMS can provide mobility services of higher quality than Mobile-IP, we implemented ISMS and Mobile IPv6 in our experimental system and measured mobility control and user packet processing performance. Then, we conducted a comparative evaluation and provided an analysis. Fig. 4 shows the ISMS prototype system configuration.

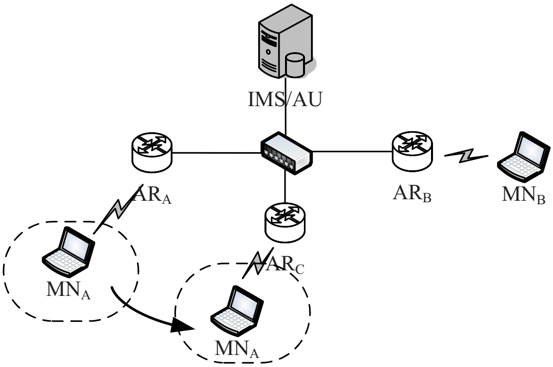


Fig. 4. Prototype system for ISMS-MANET

The software installed in the ARs and IMS/AC is based on Linux 2.6.9. The Linux kernel is modified to support the ISMS, the user level program is also developed to control the kernel and process the protocol messages.

The AR achieves the function of <AID, RID> assignation, report, query, identifiers swapping and packet forwarding. The software implemented on AR is divided into the kernel part and user land part. The kernel part has a cache of CMT/LMT and performs identifiers swapping, whereas the user layer part processes the protocol messages from MN/IMS/AC and controls the CMT/LMT. The IMS function is achieved by the user level program and managed all the <AID, RID> pairs in the network. The AC function is also achieved by user level program and manages the authentication information of all the terminals.

5.1 Performance Analysis

Firstly, we evaluate the mobility control performance of ISMS compared to Mobile IPv6 by single processing time of Location Registration and Handover.

Table 1 shows the processing delay of Location Registration and Handover in the scenario where only one MN in the network. These results do not include movement detection time, as it takes the same time for Mobile IPv6 and ISMS.

For ISMS, when a MN moves, the Handover is from step (2) to step (5) in Fig. 1, and the Location Registration is from step (4) to step (5).

Table 1. Control Procedure Delay

	Location Registration	Handover
ISMS	5.19ms	8.84ms
Mobile IP	286.41ms	1377.24ms

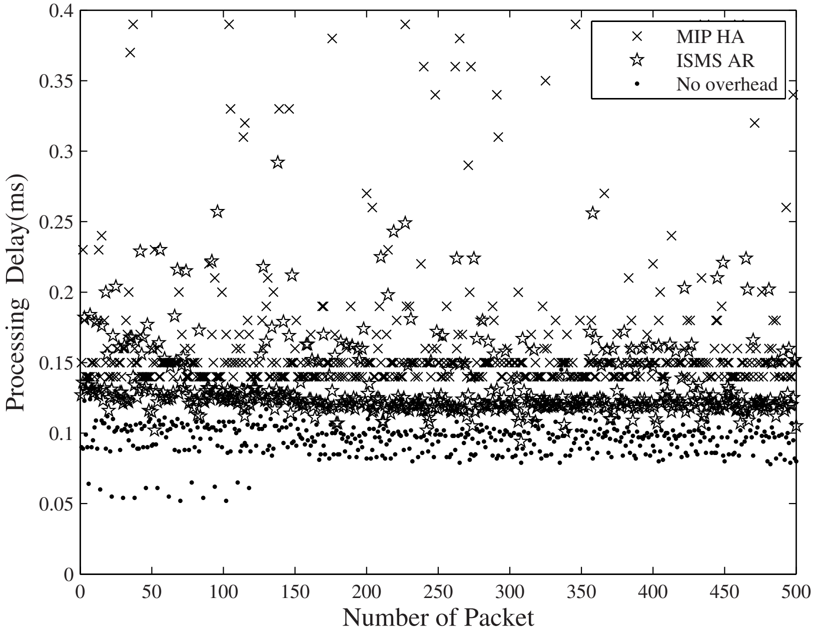


Fig. 5. User Packet Processing Delay

For Mobile IPv6, we pick up the Handover section from the MN receiving Router Advertisement, creating Care of Address (COA), performing DAD, to receiving Binding Acknowledgement. The Location Registration is from the MN sending Binding Updates to receiving Binding Acknowledgement.

From the results, we can see that the ISMS Location Registration and Handover are considerably faster than that of Mobile IPv6. This is mainly because, in Mobile IPv6 Handover, the MN has to get a new COA and performs DAD which takes more than 1000ms, whereas ISMS does not need DAD since the MN uses a permanently allocated AID regardless of its location. The complicated processing of Mobile IPv6 also takes much more time than that of the ISMS.

Secondly, we measure the single user packet processing delay to evaluate how the identifiers swapping method employed in the ISMS design influences its performance. The packet length of a user packet is 60 bytes excluding IP header, and we send the packet for 500 times. In Mobile IPv6, packet processing delay on HA is measured. Fig. 5 shows the measurement results of ISMS and Mobile IPv6.

From the results we can observe the packet processing delay at HA is relatively larger than that of the AR in ISMS. From this result we can see that, although the AR in ISMS swaps the addresses from AID (RID) to RID (AID), the load is considerably small. This evaluation shows that ISMS is effective in reducing processing delay by avoiding the increase of header length, and the identifiers swapping method employed in the ISMS is a lightweight procedure compared to Mobile IPv6 packet routing method.

6 Conclusion and Future Work

This paper proposes an Internet access solution for MANET, called ISMS-MANET. The ISMS is a proposed next generation architecture which can easily support network based multi-homing and mobility. In the ISMS, the AID represents host's identity while the RID represents the host's topological location. The RID and <AID, RID> mapping pair of a host is assigned and managed by the network instead of the host itself, which is of benefit to network management and host's the location privacy. In the ISMS-MANET architecture, the AR will implement the AODV protocol and work as a MANET node. When an AR received an AODV RREQ message, it will acknowledge the originate node by AODV RREP message and a new flag. Then the originate node starts the register procedure and establishes a new route entry towards the destination node. Except sending the Authentication Request message, the originate node deals with the RREP message from an AR in the same way as the RREP message from any node in the MANET. The solution has been evaluated by an implementation of prototype system, results show that ISMS-MANET is more effective in reducing handover and processing delay than the Mobile-IP based solution.

References

1. Perkins, C., Belding-Royer, E., Das, S.: Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561 (2003)
2. Clausen, T., Jacquet, P.: Optimized Link State Routing Protocol (OLSR). RFC 3626 (2003)
3. Johnson, D., Hu, Y., Maltz, D.: The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4. RFC 4728 (2007)
4. Johnsson, U., Alriksson, F., Larsson, T., Johannson, P., Maquire Jr, G.: MIP-MANET - Mobile IP for Mobile Ad hoc Networks. In: Mobihoc. Proceedings of First Annual Workshop on Mobile Ad Hoc Networking and Computing, pp. 75–85. IEEE Press, Boston (2000)
5. Nilsson, A., Perkins, C.E., Tuominen, A.J.: AODV and IPv6 Internet Access for Ad hoc Networks. ACM SIGMOBILE Mobile Computing and Communications Review 6(3), 102–103 (2002)
6. Benzaid, M., Minet, P., Agha, K.A.: Integration of Mobile-IP and OLSR for a Universal Mobility. Wireless Networks archive 10(4), 377–388 (2004)
7. Perkins, C.: IP Mobility Support for IPv4, revised. draft-ietf-mip4-rfc3344bis-03 (2007)
8. Johnson, D., Perkins, C., Arkko, J.: Mobility Support in IPv6. RFC 3775 (2004)
9. Lee, D., Yoo, J., Kim, K.: IPv6 Stateless Address Auto-configuration in Mobile Ad-hoc Network (T-DAD) and Performance Evaluation. In: PE-WASUN 2005, Montreal, Quebec, Canada, pp. 271–272 (2005)
10. Engelstad, P., Egeland, G.: NAT-based Internet Connectivity for On Demand MANETs. In: Battiti, R., Conti, M., Cigno, R.L. (eds.) WONS 2004. LNCS, vol. 2928, pp. 342–356. Springer, Heidelberg (2004)

11. Engelstad, P., Egeland, G., Van Thanh, D.: Analysis of NAT-Based Internet Connectivity for Multi-Homed On-Demand Ad Hoc Networks. In: CNDS 2004. Proceedings of Western Simulation Multi-conferences, Symposium on Computer Networks and Distributed Systems, San Diego, California (2004)
12. Ping, D., Ya-juan, Q., Hong-ke, Z.: Research on Universal Network Supporting Pervasive Services. Acta Electronica Sinica 35(4), 599–606 (2007)
13. Moskowitz, R.: Host Identity Protocol Implementation, Internet Draft, draft-moskowitz-hipimpl-01.txt (2001)

Using Packet Combination in Multi-query Optimization for Data Collection in Sensor Networks

Jun-Zhao Sun

Academy of Finland
Department of Electrical and Information Engineering, University of Oulu
P.O. Box 4500, FIN-90014 University of Oulu, Finland
junzhao.sun@ee.oulu.fi

Abstract. In sensor networks, queries need to be jointly designed, in order to minimize the power consumption and maximize the lifetime. Data reduction techniques can be employed to decrease the size of data to be transferred in the network, and therefore save energy of sensor nodes. This paper presents a novel method for optimizing multi-query in sensor networks. Our approach is, by using packet combination techniques, to reduce the data size of multiple simultaneous queries, so that the energy for data transmission can be saved to the best extent. A delay item is specified together with the query by the application. Then an optimal query plan can be obtained by studying the best time of sending local data to sink that can lead to the minimum cost. Algorithm is described in detail. Performance analysis is performed to validate the effectiveness of the proposed method.

Keywords: Sensor networks, multi-query optimization, delay-aware, data collection, energy-efficient.

1 Introduction

Sensor networks represent significant improvement over traditional sensors in many ways [1, 2]. However, sensor nodes have very limited supply of energy, and should be available in function for extremely long time (e.g. a couple of years) without being recharged. Therefore, energy conservation needs to be one key consideration in the design of the system and applications. Extensive research work has been devoted to address the problem of energy conservation. Examples include energy efficient MAC protocol [3], clustering [4], localization [5], routing [6], data management [7], applications [8], etc.

A sensor field is like a database with dynamic, distributed, and unreliable data across geographically dispersed nodes from the environment. These features render the database view [9-11] more challenging, particularly for applications with the low-latency, real-time, and high-reliability requirements. Under a database view, a wireless sensor network is treated as a virtual relational table, with one column per attribute and one row per data entry.

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 645–656, 2007.

Sensor network applications use queries to retrieve data from the networks. The result is a logical sub-table of the whole virtual table of the network, with each data entry an associated timestamp to denote the time of measurement. The real data table of a node is different from the one in the virtual table in the sense that there will be only one attribute there.

Query processing is employed to retrieve sensor data from the network [12-14]. A general scenario of querying sensor network is, when user requires some information, he or she specifies queries through an interface at sink (also known as gateway, base station, etc.). Then, queries are parsed and query plans are made. After that, queries are injected into the network for dissemination. One query may eventually be distributed to only a small set of sensor nodes for processing. The end nodes then execute the query by sampling phenomena/object. When sensor node has the sampling data ready, results flow up out of the network to the sink. The data can then be stored for further analysis and/or visualized for end user. Query optimization can be made through out the process in all the stages.

This paper presents a novel query optimization method for wireless sensor networks, and in particular, for the last stage of query processing: query result collection. The proposed method is applicable for the optimization of both one-time query and periodical query, for both single sink and multi-sink networks. The key novelty of the method lies on the careful consideration of timing issue along with energy consumption in data communication. By taking advantage of the delay constraint specified with a query, the method can find the optimal combination of transmitting sensor data to sink, with or without data processing. The paper takes into account the situation where multiple queries are simultaneously under processing in one single sensor networks.

The remainder of this paper is organized as follows. Section II discusses multi-query in sensor networks. Section III defines relationships for multi-query. Section IV describes the multi-query optimization method in detail. Performance of the proposed method is evaluated in Section V. Finally, Section VI concludes the paper.

2 Multi-query in Sensor Networks

2.1 Query and Multi-query

This paper focuses on the last query stage: query execution for data collection at both end nodes and reply nodes. Query execution consists of a simple sequence of operations at each node during every epoch: first, nodes sleep for most of an epoch; then they wake, sample sensors and apply operators to data generated locally and received from neighbors, and then deliver results to their parent. Nodes sleep for as much of each epoch as possible to minimize power consumption. They wake up only to sample sensors and relay and deliver results. Because nodes are time synchronized, they all sleep and wake up at the same time, ensuring that results will not be lost as a result of a parent sleeping when a child tries to propagate a message. [12]

Query processing is performed at different levels. In this paper, four levels of query can be identified, i.e. sink→query→attribute→sample, as follows.

1) Queries from different sinks. Sink serves as the point of accessing a sensor network. One sensor network can have multiple sinks that are located at different places serving for different purposes/users, as shown in Figure 1. Each sink can pour out its own query flow in to the network.

2) Queries from one single sink. Usually, one sensor network should allow multiple co-existing queries in execution simultaneously. These queries may belong to one or several applications.

3) Different attributes of one query. On the one hand, it is common that one query concerns the retrieval of multiple data attributes (corresponding to multiple columns of the virtual table of sensor network database). On the other hand, modern sensor node is usually equipped with multiple sensors for different attributes.

4) Different samples of one attribute. A query may request the collection of multiple measurements of the targeting attribute in one single query execution. These measurements may of the samples at one node (e.g. as in periodical query) or different nodes.

This paper considers the optimization of multiple queries at one single end node, and therefore covers all the four query levels above. Figure 2 illustrated the packet format designed for this paper. The format is specified quite common in the sense that all the necessary fields are presented, while ensuring only necessary fields presented as well. Note that the format can be simplified by omitting some fields, in case some of levels do not exist. Also note that reading may include a embedded time-stamp and/or node ID besides the measurement.

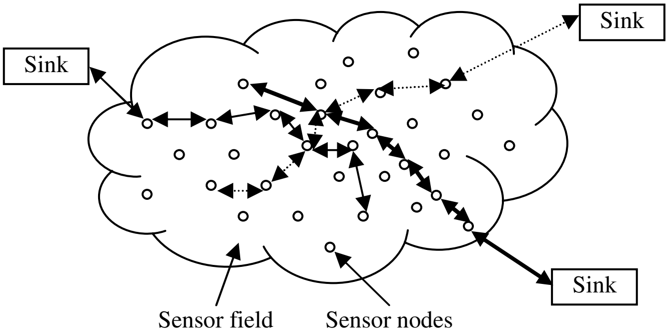


Fig. 1. Multi-sink sensor network

2.2 Query Components

A query contains five basic elements: data attributes, operations on the collected data, sensor selection predicate, QoS constraints, and temporal information. SQL-based query language, which consists of a SELECT-FROM-WHERE clause, representation is commonly accepted. However, sensor network has its own characteristics, and therefore extensions must be made to the basic SQL query. Query representation for sensor networks is out of the scope of this paper. Below is a simple example query in the form of extended SQL.

```
SELECT            temperature, light
FROM              sensors AS s
WHERE             s.location = Area_C
WHILE             delay = 10 min
SAMPLE
    ON            Now + 5 min
    INTERVAL      1 s
    LOOP          120
```

With respect to various temporal features, a query can be either *one-time query* or *continuous query*. One-time query is to sense the object only once at a specific moment of time, and is used to take a snapshot to the phenomena under observation. In one-time query, there is no INTERVAL and LOOP fields. As for continuous query, data sampling is conducted periodically for some times specified by LOOP component, with the frequency specified by INTERVAL component. Sometimes LOOP field can be replaced by UNTIL clause to give the stop criteria, in case long-term continuous sampling is needed. There is also random query which collects multiple samples in one query, but sampling interval is random instead of fixed. Periodical query can be considered as a special case of random query. Aggregate query is to summarize a set of sensor into a single statistic, like MAX, MIN, AVERAGE, etc.

This paper concentrates on the optimization of both snapshot and periodical queries during the stage of query execution for result collection. In particular, we mainly consider the situation where there is a delay constraint, and no operations (aggregation, fusion) to be performed for collected data in the query. This sort of query is very popular in real world applications like environment monitoring and healthcare.

1) Specifying delay item. Delay can be specified either absolutely, like delay = 05:30:00 or delay = Now + 10 h, meaning that the report of the query result should not be latter than a specific time, or relatively, like delay = 10 m, meaning that the report time should be with a time period after the sampling.

2) Specifying ON item. ON item denotes the sampling time. It can be a point of time, like ON 12:10:00 or ON Now + 2 h, or a period of time, like ON (07:00:00, 08:00:00) or ON 07:30:00±30m, meaning that the measurement can be conducted at any moment within the period.

3) Specifying random sampling. Random sampling can be specified by giving a period-based specification of an ON item plus a LOOP item, but without INTERVAL item, meaning that a number of samples should be collected randomly within a period of time.

3 Multi-query Relationship

This section defines different relationships between multiple queries. Suitable optimization approach can be employed by analyzing various relationships.

Obviously, periodical query and random query can be easily translated into a series of snapshot queries. Therefore without losing any generality, here we suppose that there are two snapshot queries, Q1 and Q2 under study. First, they both have an ON item, as *ON (Ts1/2, Te1/2)* (i.e. starting time and end time). Next, they both have a delay item, as *delay = Td1/2*. Here delay is given by absolute time point, which can be easily calculated from a relative delay if presented as so. Then, we can have the following definitions. First, by analyzing sample time, we have

Definition 1. *The two queries Q1 and Q2 are sample-overlap if they are for the same data attribute, and Ts1≤Te2 or Ts2≤Te1, denoted by Q1 s∩ Q2 ≠ ∅, where s∩ is sample-intersect.*

An extreme situation of Definition 1 is given below.

Definition 1.1. *Q2 is sample-covered by Q1, or Q1 sample-covers Q2, if they are for the same data attribute, and Ts1≤Ts2 and Te2≤Te1, denoted by Q2 s⊆ Q1.*

If two queries are sample-overlap, optimization can be performed so that, by choosing one common time point for sampling, the measurement can be shared by the two queries. This reduce one sample processing and thus save the power consumption. Note that Q1 and Q2 are not necessarily from the same sink. After this analysis, time of sampling for every query can be selected. We denote the selected sampling time by *T1* and *T2* for Q1 and Q2 respectively.

Next, by analyzing delay time, we have the following definitions.

Definition 2. *The two queries Q1 and Q2 are delay-overlap if they are from the same sink, and MAX (T1, T2) ≤ MIN (Td1, Td2), denoted by Q1 d∩ Q2 ≠ ∅, where d∩ is delay-intersect.*

Similarly, an extreme situation of Definition 2 is given by:

Definition 2.1. *Q2 is delay-covered by Q1, or Q1 delay-covers Q2, if they are from the same sink, and T1≤T2 and Td2≤Td1, denoted by Q2 d⊆ Q1.*

If two queries are delay-overlap, optimization can be performed so that, by merging the readings of the two queries into one packet, the transmission power may be saved by reducing the total size of data volume to be sent. Note that Q1 and Q2 are not necessarily for the same data attribute.

Definition 2 and 2.1 are defined for end nodes where the queries are executed, however can be easily extended to the case of median nodes (i.e. nodes for replying messages from end nodes to sinks). In this case, either Q1 or Q2, or both of them are executed at an offspring node of local node. Therefore, the sampling time T1 or T2 must be replace by the time when the packet is received.

Based on the definitions above, it is clear that by identifying the existence of sample-overlap and delay-overlap, query execution may be optimized with respect to energy consumption in both data processing (in case of sample-overlap) and data transmission (as of delay-overlap). This is the key idea of the paper. However, since optimization operations bring time and energy overhead as well, we need to build up model to exam the effectiveness of the possible optimization, in terms of the overall energy consumption of the whole sensor network, before making a decision whether the optimization should really be performed or not.

4 Multi-query Optimization

4.1 Packet Combination

Data reduction is to decrease the size of data that is needed in the communication. The idea is straightforward: less amount of data consumes less amount of power in transmission. Various data reduction techniques exists in this context, including packet combination, packet compression, data aggregation, and data fusion. This paper only studies the first technique: packet combination.

Packet combination is a simple data reduction technique, which combines multiple small packets into a big one, without considering the correlations between and the semantics within individual packets. In wireless communication, it is much expensive to send multiple smaller packets instead of one larger packet. One packet contains two parts, header and payload. Packet header is the packet overhead whose format is common for all the packet, which contains numbering, addressing and error checking information, as shown in Figure 2. This commonly used packet structure forms the basis of packet merging – multiple packet can be combined so that only one packet header is presented with the rest the combination of the payloads of all the packets.

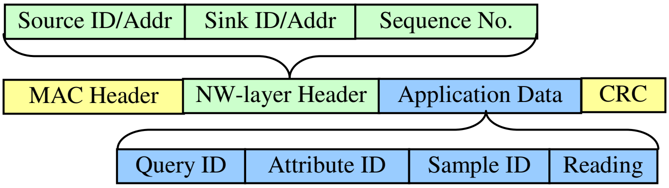


Fig. 2. Packet format

In the consideration of the four query levels defined in Section II.A, corresponding packet combination techniques can be employed. Figure 3 illustrated the packet formats resulted from packet combinations at different levels. All the levels of combinations can coexist, and some levels may be omitted in real applications.

Packet combination is aiming at reducing the total data size in transmission. Data reduction ratio, *dr* can be defined as

$$dr = 1 - D'(u)/D(u), \tag{1}$$

where $D(u)$ is the size of data before the packet combination, and $D'(u)$ is the size after, $D'(u) \le D(u)$. Obviously, a higher *dr* is expected. The real value of *ru* is mostly depending on the number of features to be combined. In this paper, it is reasonable to assume that $0 \le dr < 1$ always holds.

Finally, concerning packet combination at level 1 – sink level, the combination can be performed only when, in all the routing trees with the roots the sinks to be combined, current node has a common parent node.

4.2 System Model

In a sensor network, there are both energy and time consumptions resulted from both computation and communication. The computation concerns data sampling by sensors, as well as local and in-network data processing like packet combination. Communication component allows a set of spatially distributed sensor nodes to send data to a central destination node. Therefore, the power consumption of the sensor network consists of two types of energy cost, data processing cost and data transmission cost. Similarly, the time consumption of the sensor network consists of data processing time and data transmission time.

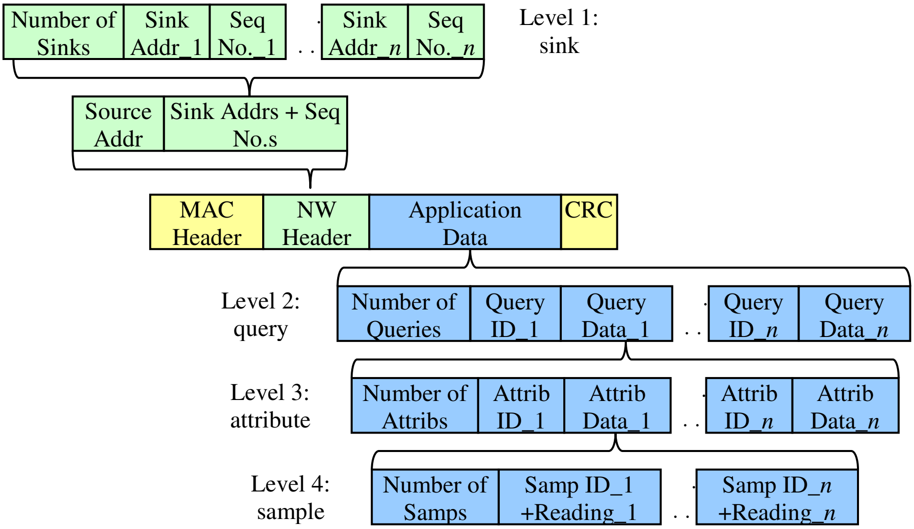


Fig. 3. Packet combination

According to [15], transmitting one bit via radio medium is at least as 480 times power consuming as performing one addition instruction. Moreover, techniques like compression can also be used to each individual packet even if there is no optimization performed over multiple packets, which power consumption counts as well. Furthermore, packet combination costs energy mostly because of memory R/W operations, while in normal situation (i.e. no packet combination) there are still such memory operations. On the other hand, data transmission in sensor networks is multi-hop by nature, meaning that multiple median nodes are involved in the transmission. Therefore, the energy cost of packet combination at end node is tiny comparing to the total energy consumption resulted from multi-hop transmission. Based on all the discussions above, it is reasonable to ignore energy and time costs that are resulted from packet combination in this paper. It is worthy noting that it is very easy to extend the following idea of communication models creation to build models for packet processing models.

A sensor network is modelled as a graph $G = (V, E)$, where V denotes the node set and E the edge set representing the communication links between node-pairs. An edge $e \in E$ is denoted by $e = (u, v)$, where u is the start node and v is the end node.

Transmission cost denotes the cost for transmitting $D(u)$ amount of data (i.e. packet header plus payload) from node u to node v through link $e = (u, v)$. The cost includes the energy consumption at both u and v. Unit cost of the link for transmitting data between two nodes can be abstracted as $C_U(e)$, and thus the transmission cost $C_T(e)$ is given by

$$C_T(e) = C_U(e)\, D(u), \tag{2}$$

The unit transmission cost on each edge, $C_U(e)$, can be instantiated using the first order radio model presented in [16]. According to this model, the transmission cost for sending one bit from one node to another that is d distance away is given by $\beta d^{\gamma} + \varepsilon$ when $d < r_c$, where r_c is the maximal communication radius of a sensor, i.e. if and only if two sensor nodes are within r_c, there exists a communication link between them or an edge in graph G; γ and β are tunable parameters based on the radio propagation, and ε denotes energy consumption per bit on the transmitter circuit and receiver circuit.

Similarly, transmission time $T(e)$ is given by

$$T(e) = T_U(e)\, D(u), \tag{3}$$

where $T_U(e)$ is the unit transmission time, i.e. the reciprocal of bandwidth, whose value is depended on the condition of the link. We note that all the cost and time parameters are all defined on link e, because different link has different conditions e.g. distances, congestion, and reliability.

The model above can be easily extended to the transmission cost and time for a *path*, which are the ones utilized in this paper. The cost and time for $D(u)$ from one node x to another node y through a multihop path $x\text{->}y$ can be represented as $C(D(x)\text{: } x\text{->}y) = \sum_{e \in x\text{->}y} C(e)$ and $T(D(x)\text{: } x\text{->}y) = \sum_{e \in x\text{->}y} T(e)$. Therefore, total energy saving, ES is given by

$$ES = C(D(x)\text{: } x\text{->}y) - C(D'(x)\text{: } x\text{->}y) = dr\, D(x) \sum_{e \in x\text{->}y} C_U(e) \tag{4}$$

It is worthy noting that in the following study the destination node y is always the single sink node (denoted by s hereafter).

4.3 Algorithm

To derive the algorithms for energy-efficient data collection based on packet combination, there are following hypothesises defined in this paper. We assume that the topology of the sensor network is organized as a spanning tree, with the sink as the root, and tree-based routing algorithm is adopted. In case of node mobility or link failures, the routing tree will be automatically re-configured by the underlying topology control protocols. Downlink (from sink to leaf nodes) and reverse uplink are symmetrical, meaning that the communication bandwidth is at the same level. A global synchronization is achieved in the network, that is, all the nodes share a common clock. There exists a restriction on the max packet size (MPZ) allowed for transmission in the sensor network. Each sensor node has full knowledge of local information, including local routing information (i.e. maintain a record of its parent nodes and all the next level offspring nodes) and the queries that are on its branch.

Before data collection, in order to support the proposed optimization algorithms, there must be some pre-processing during query dissemination stage. Queries are injected from the sink into the sensor network for dissemination. On receiving a query, a node stores the query ID and the query itself (attributes, delay, sampling time, interval, loop, etc.). In other words, each node maintains a list of valid queries. The stored query entry becomes expired when execution is completed (in the case when the query is for this node, according to the *where* clause.) and the latest time to report result passes (no matter whether the query is or is not for this node).

The core of the optimization algorithm is to make a decision about to a set of queries at local node, whether to perform the optimization, and if yes, when and to whom. Following in the algorithm, we assume node *u* is under investigation. The operation below is carried out each time when the node wakes up.

a. When receiving a query, the node checks the query list to see if there are any other queries. If no, the query is put into the list. For each attribute/sample, set *sampling interval* to (*Ts, Te*) and *sampling time* to (*Ts*+*Te*)/2, i.e. the middle point of the sampling interval; also set the *lastest time of transmission* to *delay-T*(*MPS: u->s*), i.e. the time before which the combined packet must be sent.

b. When receiving a query, if there are already some queries in the query list, then check each sample of the current query with each of those in the list. If there is *sample-cover*, then mark one of them as "*covered*", adjust the *sampling interva*l to (*MAX(Ts), MIN(Te)*), and adjust the *sampling time* according to the new interval. Check all the other samples and adjust the *sampling interval* and *sampling time* if necessary.

c. When receiving a (combined or not combined) packet from a offspring node of node *u*, restore all the samples from the packet, mark them "*done*", and set *latest time of transmission* according to the stored query information.

d. If to a sample of a query in the list, it is the *sampling time*, then the node samples and stores the measurement, and mark this and all the samples covered by this sample "*done*".

e. If otherwise to a sample of a query in the list, it is the *latest time of transmission*, then try to combine all the samples that are marked as "*done*" according to the method introduced in Section IV.A and Figure 3, until MPS achieves. Those samples can be combined with the current one, meaning that they *delay-cover* it.

5 Performance Analysis

This section analyses the performance of the proposed query optimization method. Table I lists the parameters and the values used in the following experiments.

Table 1. Parameters values used in performance analysis

parameter	value
β	100 pJ/bit/m^2
γ	2
ε	50 nJ/bit
d	1 – 30 m
MAC Header	5 bytes
CRC, Sink Addr, Sequence No.	2 bytes
Number of Sinks/Queries/Attributes/Samples	1 bytes
Sink/Query/Attribute/Sample ID	1 bytes
MPS	2k bytes

First we study data reduction ratio *dr* with respect to size of reading. In this experiment, we assume a packet combination performed above 3 queries, with 2 attributes per query. Numbers of samples per attribute are set to 1, 5, and 10. Size of readings are ranging from 2 to 100 bytes. Figure 4 shows the results of this experiment. It can be seen in the figure that data reduction ratio decreases with the increasing of the size of readings. This is because the sizes of the headers of a packet plays less important role when increasing the size of the readings.

Number of samples influences the total samples to be combined into one or several packet, and thus is significant to *dr* as well. Obviously, more combined samples leads to higher *dr*. However, when the number of samples increases to a certain degree, the difference between the corresponding ratios is tiny. This is due to the same reason above. Packet combination lies on sharing packet headers. Therefore, the more the headers comparing to sampling data, the better the reduction ratio. However, this does not mean that in practice less samples with small sizes are privilege to conduct packet combination.

Second experiment is conducted to study the impact of distance between to nodes to the energy cost in one transmission of packet with/without combination. To simplify the study, we set the original packet size to MPS, and choose three cases of

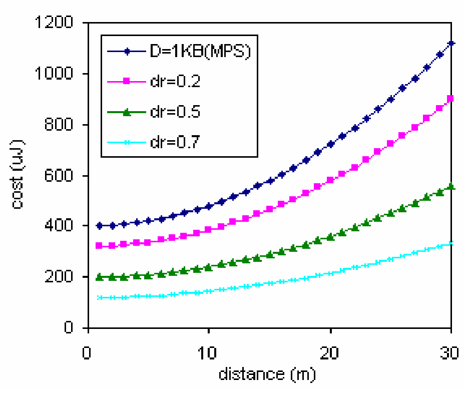


Fig. 4. Impact of size of readings to data reduction ratio

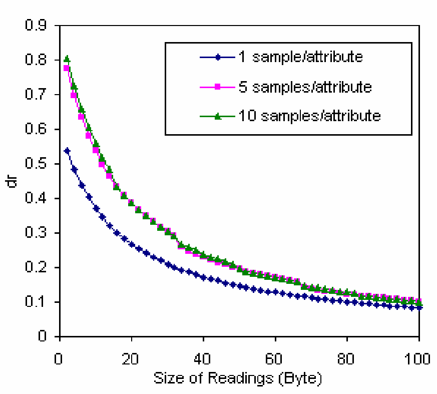


Fig. 5. Impact of distance to energy consumption in single hop

combination, in which the data reduction ratios vary as 0.2, 0.5, and 0.7. Figure 5 shows the results. It is clear that distance between nodes heavily affects the energy consumption in data transmission. The far the distance, the more power needed. It can also be seen from the figure that cost dramatically decreases with high reduction ratio. Figure 6 shows the energy saving in the same setup, with each comparing to the original situation where there is no optimization performed.

The last experiment is conducted aiming at studying the overall effect of the optimization method to the whole network. The target sensor network is modeled as a 7×7 grid. Nodes are at the points of intersection with distance of 1m, and sink is at left-up corner. Therefore, there are totally 49 sensor nodes. We assume that each node produces one sampling data the size of which ranges from 1 to 20 bytes. The sink then collects all the data, during the process optimization may performed (with-op) or not performed (no-op). All nodes act as both routers and sources. When optimization is performed, median node combines local reading with the data received from all the offspring nodes before transmitting to its parent node. Figure 7 shows the total cost of energy of the network (i.e. the sum of energy consumptions of all the nodes). It is clear that data collection with proposed optimization greatly reduces the total power consumption of the whole network.

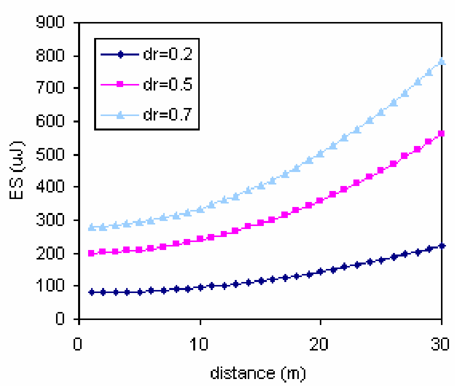


Fig. 6. Impact of distance to energy saving in single hop

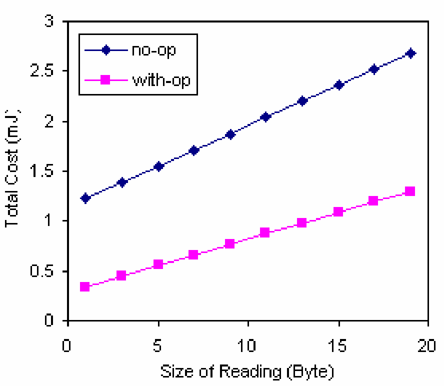


Fig.7. Impact of size of reading to energy consumption in multi-hop

6 Conclusion and Future Work

A packet combination based method is proposed to optimize the execution of multi-query in a sensor network, by considering the costs of energy with the application-specific delay. Algorithm is described in detail. Experiments are conducted to validate the method. Results show that the proposed method can achieve the goal of query optimization. Communication links can be unreliable, and in this case long packet may have high probability to incur errors during the transmission. Also, similar model can be easily created for considering the time and energy costs resulted from packet combination operations. These two issues are the problems for future investigation.

Acknowledgment. Financial support by Academy of Finland (Project No. 209570) is gratefully acknowledged.

References

1. Gehrke, J., Liu, L.: Sensor-network applications. IEEE Internet Computing 10(2), 16–17 (2006)
2. Gharavi, H., Kumar, S.P.: Special Issue on Sensor Networks and Applications. Proceedings of the IEEE 91(8) (August 2003)
3. Miller, M.J., Vaidya, N.H.: A MAC Protocol to Reduce Sensor Network Energy Consumption Using a Wakeup Radio. IEEE Transactions on Mobile Computing 4(3), 228–242 (2005)
4. Fukushima, Y., Harai, H., Arakawa, S., Murata, M.: Distributed clustering method for large-scaled wavelength routed networks. In: Proc. Workshop on High Performance Switching and Routing, pp. 416–420 (May 2005)
5. Hu, L., Evans, D.: Localization for Mobile Sensor Networks. In: MobiCom 2004. Tenth Annual International Conference on Mobile Computing and Networking, Philadelphia, pp. 45–57 (September-October 2004)
6. Al-Karaki, J.N., Kamal, A.E.: Routing techniques in wireless sensor networks: a survey. IEEE Wireless Communications 11(6), 6–28 (2004)
7. Demers, A., Gehrke, J., Rajaraman, R., Trigoni, N., Yao, Y.: Energy-Efficient Data Management for Sensor Networks: A Work-In-Progress Report. 2nd IEEE Upstate New York Workshop on Sensor Networks. Syracuse, NY (October 2003)
8. Zou, Y., Chakrabarty, K.: Energy-Aware Target Localization in Wireless Sensor Networks. In: PerCom 2003. Proc. 1st IEEE International Conference on Pervasive Computing and Communications, Dallas-Fort Worth, Texas, USA, pp. 60–67 (March 2003)
9. Govindan, R., Hellerstein, J.M., Hong, W., Madden, S., Franklin, M., Shenker, S.: The sensor network as a database, USC Technical Report No. 02-771 (September 2002)
10. Bonnet, P., Gehrke, J.E., Seshadri, P.: Towards Sensor Database Systems. In: Proceedings of the Second International Conference on Mobile Data Management, Hong Kong (January 2001)
11. Madden, S.R., Franklin, M.J., Hellerstein, J.M., Hong, W.: TinyDB: An Acquisitional Query Processing System for Sensor Networks. ACM Transactions on Database Systems 30(1), 122–173 (2005)
12. Madden, S., Franklin, M.J., Hellerstien, J.M., Hong, W.: The design of an acquisitional query processor for sensor networks. In: Proceedings ACM SIGMOD, San Diego, CA, USA, pp. 491–502 (June 2003)
13. Gehrke, J., Madden, S.: Query processing in sensor networks. IEEE Pervasive Computing 3(11), 46–55 (2004)
14. Yao, Y., Gehrke, J.: Query processing for sensor networks. In: CIDR 2003. Proceedings of the First Biennial Conference on Innovative Data Systems Research, Asilomar, California (January 2003)
15. Kimura, N., Latifi, S.: A survey on data compression in wireless sensor networks. In: ITCC 2005. Proceedings of the International Conference on Information Technology: Coding and Computing, vol. 2, pp. 8–13 (2005)
16. Heinzelman, W.R., Chandrakasan, A., Blakrishnan, H.: Energy-Efficient Communication Protocol for Wireless Microsensor Networks. In: Proc. 33rd Ann. Hawaii Int’l Conf. System Sciences (January 2000)

Relative Positions Within Small Teams of Mobile Units

Hongbin Li[1], Luis Almeida[2], Zhi Wang[1,⋆], and Youxian Sun[1]

[1] State Key Laboratory of Industry Control Technology, College of Info Science and
Engineering, Zhejiang University, Hangzhou 310027, P.R. China
{hbli, wangzhi, yxsun}@iipc.zju.edu.cn
[2] IEETA-DETI, University of Aveiro, 3810-193 Aveiro, Portugal
lda@det.ua.pt

Abstract. It is common that a small group of autonomous mobile units requires location information of each other, but it is often not applicable to build infrastructure for location service. Hence, mobile units use RF signals to determine their relative positions within the group. Several techniques have been proposed for localization, but none of them consider both units mobility and anchor unavailability. In this paper we develop a propagation scheme for spreading the signal strength information through the network, and filtering techniques are employed to process the noisy signal. We use Floyd-Warshall algorithm to generate pairwise signal distance of each pair of units. Then Multidimensional Scaling technique is used to generate relative position from pairwise distances. Due to anchor unavailability, relative positions are adjusted by certain rules to reflect the continuous mobility. We verify our methods in MICAz platform. Experimental results show that we can obtain smooth relative positions under mobility and manage moving patterns of mobile units.

Keywords: Mobile Robot, Relative Position, Radio Signal Strength.

1 Introduction

It is attractive to see a team of mobile robots cooperating with each other for a common goal, with no human intervention. Sample applications include surveillance, environment exploration, and manufacturing.

Consider the scenario of home intelligent robots, each of them is equipped with identical basic robotic component and communication module, so that they can move in a common pattern and communicate in a predefined channel. And for the sake of cost, robots have different sensing or actuating components, which means some tasks have to be accomplished by specific robots. When one robot detects a special event, it may have to notify another robot, which is far from the event area but equipped with specific actuating component, to deal with such event.

In the application of mine sweeping, it is wise to spread a team of robots with mine detecting capability and equip only a small portion of them with sweeping ability. A typical scenario is depicted in Figure 1. The mine field is divided into small areas according to the robots' sensing range. Robots sweep the areas one by one with certain

⋆ Corresponding author.

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 657–671, 2007.

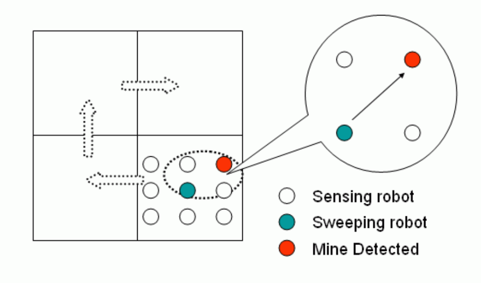


Fig. 1. A scenario of mine sweeping

formation to guarantee coverage. When mines are detected, robot with sweeping ability is informed to approach the specific area. For both formation maintaining and sweeping robot relocating, robots have to manage their relative locations.

In both cases, it is important to notify a robot to move from one area to another. For home robots, a possible solution is to build an infrastructure that enables every robot to know its own absolute position. But building infrastructure is costly and it is probably unavailable in urgent scenarios. For mine sweeping, GPS may be a possible solution, but it is satellites dependent and only provides coarse-grained positions. A more elegant solution is to derive relative positions through local communication.

In this paper, we use RF signal strength as a measure of distance, despite the coarse relationship between both. Therefore, we use the concept of relative position in signal strength space and we apply it to a small team of mobile units without anchor information. We first design a scheme for managing the RF signal strength sensed in each unit via the propagation of a connectivity matrix. Then we calculate the relative position of the mobile units and plan for the movement. Experiments show that, with the relative positions in signal strength space, we can help wireless connected units manage the relative positions without central supervision. Mind that as we study the relative positions in signal strength space, the physical accuracy is not a major concern.

2 Related Work

Lots of effort has been devoted to robots formation and path planning. [1] explores the idea of using feedback laws to control multiple robots together in a formation. They assumed that each robot has the ability to measure the relative position of other robots that are immediately adjacent to it. [2] computes robots paths to ensure network partition never occur during robots' motion, and global knowledge of location is assumed available. [3] explores sensor relocation to deal with sensor failure or respond to new events. Methods of finding redundant sensors and moving sensors to specific area are proposed in this work. They assume that sensors are placed into grids and global information is shared to support relocation planning. None of these work consider the practical position management of mobile robots or sensors.

Static sensor positioning has been widely investigated in recent years. The Received Signal Strength Indicator (RSSI), Time of Arrival (ToA)[4], Time Difference of Arrival (TDoA)[5] and Angle of Arrival (AoA)[6] are dominant methods to obtain range estimation from sensing signals. With ranging data collected, researchers developed different techniques to generate sensor positions [7][8][9][10][11]. But most work assume precise position of anchor nodes, which is unavailable or unnecessary in certain cases.

Adding mobility to sensor units appears to make localization more complex and uncertain. [12] imports Monte Carlo Localization method to improve accuracy. In their work, seeds that know their location and nodes with unknown location form a network. They investigate scenarios in which at least one kind of sensors are moving.

This paper investigates the problem in which no anchor node exists. We use Multidimensional Scaling (MDS) [7] and RF signal strength readings to compute relative positions of small teams of mobile units (approx. up to 10), and manage the mobility from generated result. To the best of our knowledge, this is the first experimental study of relative positions for mobile units with no anchor information.

The hardware platform we use in this work is Crossbow's MICAz[13] which communicates in 2.4 GHz IEEE 802.15.4. We use the RSSI and Link Quality Indicator (LQI) values measured with each packet reception to obtain the signal strength and quality of link respectively. [14] gives a comprehensive understanding of RSSI and LQI values provided by CC2420[15].

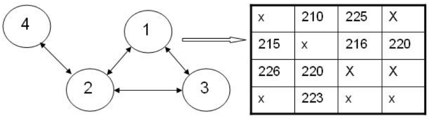
3 Connectivity Matrix and ITS Propagation

Due to high mobility together with units joining and leaving, we need a scheme to keep track of the connectivity information within the group. [16] proposed a concept called connectivity matrix, which enables every mobile unit keep a global vision of the whole team. In the connectivity matrix model, units are well synchronized and broadcast messages in a TDMA way. Each unit expects message from certain unit in the corresponding time slot, and updates the connectivity state of that unit to the local matrix according to whether the message can be received. When its own time slot comes, it broadcasts the newest matrix it has to the neighboring units. After finite period, the matrices in each unit will be identical, until topology changes again.

There are two practical limitations of the connectivity matrix propagation. First, the system must support time synchronization, which requires substantial time slot reassignments when the number of mobile units changes frequently. Second, the model uses binary representation ”1” and ”0” to denote whether the unit can successfully receive message from the corresponding unit. And this representation gives no information about how well these units are connected. When one unit is in the edge of the radio range of other, the connectivity state will probably switch between ”1” and ”0”, which causes unnecessary connectivity changes.

3.1 The Extended Connectivity Matrix

In our system model, we ignore the time synchronization. It is generally recognized that the cost of retransmissions is cheaper than the cost it would take to prevent them in lightly loaded communication conditions. MICAz communicates in 2.4G Hz with a transmission rate of 250kbps, a typical packet of 50 bytes need less than 2ms to finish. With a broadcast period of one second, the probability of channel collision for 5-10 nodes is very small.



x	210	225	X
215	x	216	220
226	220	X	X
x	223	x	x

Fig. 2. Extended connectivity matrix

As mentioned above, Chipcon 2420 provides RSSI and LQI values for each received packet. We utilize this nice property to define an extended connectivity matrix.

$$M^k(i,j) = \begin{cases} RSSI_{from\ j\ to\ i}, & LQI(receivedpacket) \geqslant LQI_{threshold} \\ 0, & \text{packet lost or } LQI(receivedpacket) < LQI_{threshold} \end{cases} \quad (1)$$

Each mobile unit maintains a connectivity matrix M. M^k is the matrix stored in unit k. For each row i, $M^k(i)$ stores the RSSI readings measured by unit i, including for $i = k$. Fig. 2 shows an example of an extended connectivity matrix with 4 nodes and their wireless links represented in the left side.

3.2 Updating Algorithm

For each round, units expect packets from each other. When packets are received with a LQI which is above a certain threshold, it is accepted and the corresponding RSSI value is put into sampling buffer. Else, the packet would be dropped. Here a sampling buffer is employed to store the latest RSSI readings and avoid sudden topology changes due to occasional packet loss. When it is time to broadcast its own connectivity matrix, the unit puts the average of non-zero RSSI readings stored in the sampling buffers to the corresponding row. The updating algorithm is described in Table 1.

Table 1. Connectivity matrix updating algorithm

```
Receive phase:
If unit $k$ receives the expected $M^w$ {
  For each $i \neq k$   $M^k(i) = M^w(i)$
  If $(LQI_{w->k} > LQI_{Threshold})$   Add($SamplesBuffer, RSSI_{w->k}$)
}
  Else
   Add($SamplesBuffer, 0$)

Broadcast phase:
If timer triggered for unit $k$ to broadcast $M^k$ {
  For each $i$ {
    $M^k(k,i)$ = Average(NonZero($SamplesBuffer(i)$))
    $M^k(i,i) = 0$
  }
  Broadcast($M^k$)
}
```

Table 2. Updating Algorithm for Connectivity Matrix with timer and sequence control

```
Receive phase:
If unit $k$ receives the expected $M^w$ {
  For each $i \neq k$ {
    If $Sequence^w(i) > Sequence^k(i)$ {
      $M^k(i) = M^w(i)$
    }
  }
  If $(LQI_{w->k} > LQI_{Threshold})$ Add($SamplesBuffer$,$RSSI_{w->k}$)
}
Else
   Add($SamplesBuffer$, 0)

Broadcast phase:
If timer triggered for unit $k$ to broadcast $M^k$ {
  For each $i$ {
    $M^k(k, i)$ = Average(NonZero($SamplesBuffer(i)$))
    If $M^k(k, i) \neq 0$
     $Time^k(i) = localtime$
    $M^k(i, i) = 0$
  }
  $Sequence^k(k) = Sequence^k(k) + 1$
  Broadcast($M^k$,$Sequence^k$)
}

Eliminate outdated RSSI phase:
If timer triggered for eliminating outdated data
  For each i {
    If $localtime - Time^k(i) > OutdateThreshold$
      $M^k(i) = \{0\}$
  }
```

Nevertheless, this algorithm is only effective to add information to the matrices, such as when new links are established. On the contrary, it is not capable of removing stale information from the matrices when a given unit crashes or leaves the team, or simply when a link is broken. In this case we need to add two control variables, a local timer that tells us how old is an RSSI value detected by that node (those kept in the unit's own line) and a sequence number that tells us, between the same line in two matrices, which is the one more up-to-date. The former is updated with a timestamp whenever a non-null value is written by a node in its own matrix (there is a vector of timers, one for each potential link). The latter is updated whenever the the node sends a new matrix and is transmitted together with it. Notice that there is also a vector of sequence numbers concerning all lines in the matrix. In the receiving phase, the sequence number allows rejecting older lines. Table 2 summarizes our improved distributed algorithm to manage the extended connectivity matrix.

4 Generating Relative Position in Signal Strength Space

Using the communication scheme of the previous section, a small group of mobile units can share topology information along with signal strength readings of each pair of units. In each unit, a coarse grain global vision can be generated locally. We consider two settings of RF transmission power: identical and different transmission power respectively. For identical transmission power, every unit has a similar transmission range and the receive power can be converted to distance according to the log-normal model or empirical data. The physical relative positions of group members can be estimated with these information. On the other hand, if units communicate in a different transmission power, which is more common in practice, the units can still generate a relative vision in signal strength space. That is, we deem units with strong RF connection to be neighboring units, and units with weak connection to be faraway from each other.

The relationship between relative positions in physical space and in signal strength space is illustrated in Figure 3. Circles denote the physical units positions and bricks denote positions in signal strength space. As we can see, the distance in signal strength space only depends on the signal strength and not directly on the physical distance.

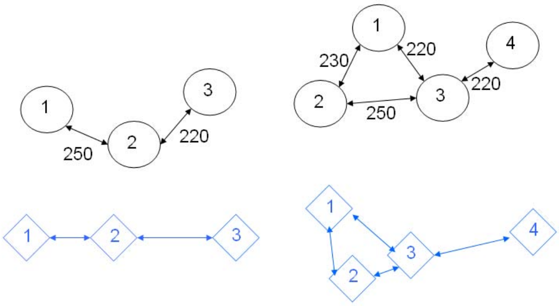
4.1 Multidimensional Scaling

We use MDS to compute the relative positions. MDS is a technique used in multivariate analysis. It transfers a known $n \times n$ matrix A of dissimilarities to n points of an m-dimensional Euclidean space in such a way that the pairwise distances between points are compatible with the dissimilarities matrix.

$Y = [y_{ij}]_{n \times m}$ denotes the physical positions of a group of mobile units. And $d_{ij}(Y)$ denotes the distance between units i and j based on their physical position Y.

$$d_{ij}(Y) = (\sum_{a=1}^{m} (y_{ia} - y_{ja})^2)^{\frac{1}{2}} \tag{2}$$

The measured distance between i and j is denoted by δ_{ij}. We assume that δ_{ij} is equal to $d_{ij}(Y)$. Let $X = [x_{ij}]_{n \times m}$ denote the estimated positions of the mobile units. And $d_{ij}(X)$



(a) A case of 3 units (b) A case of 4 units

Fig. 3. Relative positions in physical space and in signal strength space

denotes the distance between i and j based on their calculated positions. The goal of MDS is to find a X to minimize a stress function:

$$\sigma(X) = \sum_{i<j} w_{ij}(d_{ij}(X) - \delta_{ij})^2 \tag{3}$$

w_{ij} is a weighted factor that denotes the significance of the measured data. If all pairwise distances of mobile units are collected, we can use the classical MDS method to compute the positions:

1) Compute the squared distance D^2, where $D = [d_{ij}(Y)]_{n\times n}$;

2) Compute the matrix J with $J = I - e \times e^T/n$, where e = [1,1,...,1]';

3) Apply double centering to this matrix with $H = -\frac{1}{2}JD^2J$;

4) Compute the eigen-decomposition $H = UVU^T$;

5) We denote the matrix of largest i eigenvalues by V_i and U_i the first i columns of U. The coordinate matrix X is $U_iV_i^{\frac{1}{2}}$

4.2 Approximating Missing Measurements

In practice, collecting all pairwise distances is sometimes impossible. With wireless mobile units, this is the case when some links are broken due to mobility or limited communication range. Then the connectivity matrix will contain empty values (0) and the classical MDS algorithm cannot be directly applied.

[7] proposed an iterative MDS to compute the relative positions under missing information. In this approach, the weight factor w_{ij} is assigned to 1 for measured distance and 0 for missing distance. However, the stress function (3) does not put any constraint on missing distances. In other words, if unit i and unit j fail to communicate, the iterative MDS would generate a set of unit coordinates regardless of the distance between i and j. In practice, if a pair of units fail to communicate, they are probably out of the communication range of each other. In both physical space and signal strength space, non-communicating units should be considered faraway from each other.

We herein assume a fully connected network (despite possibly not fully linked), meaning that there exists at least one route between any pair of units. This also means the network is not partitioned. Let E denote a route between i and j, which contains several links, and let the pair of units a and b be the extremes of a generic link in E). We approximate the distance between two units with minimum accumulated *Signal Distance*:

$$SignalDist(i, j) = \begin{cases} RSSI_{max} - M^k(i, j), & \text{if i and j are linked} \\ min(\sum_{\forall(a,b)\in E} SignalDist(a, b)), & \text{if i and j are not linked} \end{cases} \tag{4}$$

As shown in Figure 4, the physical distance between two indirectly connected units is probably smaller than the minimum accumulated distance of a connection route. For example: $Dist_{13} < Dist12 + Dist23$. This introduces a small deformation in the nodes relative positioning but we are not aiming at determining exact physical positions but positions in the *signal strength space* which, by nature, is already not very accurate.

Table 3. Floyd-Warshall algorithm for shortest signal distance

```
(for node $k$)
For each $(i, j)$ {
    If $M^k(i, j) \neq 0$ $SignalDist(i, j) = RSSI_{max} - M^k(i, j)$
    Else $SignalDist(i, j) = Dist_{max}$
}
For $l = 1$ to $N$
    For $i = 1$ to $N$
        For $j = 1$ to $N$
            If $SignalDist(i, j) < SignalDist(i, l) + SignalDist(l, j)$
                If $M^k(i, j) = 0$
                    $SignalDist(i, j) = SignalDist(i, l) + SignalDist(l, j)$
```

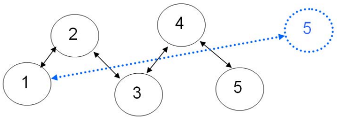


Fig. 4. Approximating the distance for missing RSSI measurement

On the other hand, this distance estimation is deterministic and easy to compute, enabling a smooth position estimation in scenarios of moving units, as we will show in the experiments section.

The classical MDS computation needs every pairwise signal distance. Table 3 describes the Floyd-Warshall algorithm for computing shortest signal distance for every pair of units. Before running this algorithm, we initially assign a maximum distance value to the initial SignalDist(i,j) in which unit i and j can not communicate directly. After the computation, a full matrix of signal distance is presented. For directly connected pairs, the distances are derived from RSSI readings. And for indirectly connected pairs, the distances are derived in a fashion of shortest path. Our distance approximation approach requires an additional time complexity of $O(n^3)$ from Flyod-Warshall algorithm. It is affordable for small teams of mobile units. And by generating the complete distance we can use classical MDS for computation, thus avoiding iterative methods.

4.3 Adjusting the Relative Coordinates

So far all we discussed the relative position of a team of mobile units with no physical anchor. However, for the MDS algorithm, a small perturbation of the distance matrix D would bring a totally different result of coordinates X. For a continuous observation of the relative position in *signal strength space*, a certain adjustment of the relative coordinates is needed. Here we only consider the result presented in 2-D space ($m = 2$).

In our adjustment, $R = [r_{ij}]_{n\times 2} = (R_1, R_2, \ldots, R_n)'$ denotes the results calculated from MDS method. $S = [s_{ij}]_{n\times 2} = (S_1, S_2, \ldots, S_n)'$ denotes the target coordinates. We consider the three units with the smallest IDs are relative anchors, which are unit 1,2 and 3 for simplicity. The coordinates adjustment usually includes shift, rotation and reflection. We place unit 1 on the origin point (0,0), unit 2 on the positive Y axis and unit 3 on the right half-plane. The rules for unit 1 to 3 correspond with shift, rotation and reflection respectively.

First we let $S_1 = (0,0)$, then compute the clockwise angle α from vector $(R_2 - R_1)$ to Y axis. An intermediate result T can be computed as in equation 5, where Q is given by equation 6. Then we check if T_3 is on the right half-plane. If so, $S = T$, else we reflect T over the vertical axis as in equation 7.

$$(T_1, T_2, T_3, \ldots, T_n) = (S_1, R_2 - R_1, R_3 - R_1, \ldots, R_n - R_1) \times Q \qquad (5)$$

$$Q = \begin{pmatrix} cos(\alpha) & sin(\alpha) \\ sin(\alpha) & cos(\alpha) \end{pmatrix} \qquad (6)$$

$$S = T \times \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \qquad (7)$$

4.4 Filtering the Noisy Signal

There are two kinds of noise in our system. One is the unstable RSSI readings. Two units, even placed in fixed position without human activity or electromagnetic interference, would receive fluctuant RSSI readings from each other. For a group of mobile units, this noise is even harder to handle.Some previous studies use techniques like averaging, frequency diversity and signal modeling to counteract the unstable signal. The second one is occasional packet loss. Due to the unreliability of wireless communication, occasional losing packets is inevitable. How to distinguish unit absence from packet loss is a critical problem for mobile units management.

For the unstable RSSI readings, we use Kalman filter[17] to counteract the unpredictable fluctuation. A one-dimensional model is employed to estimate the state $SigStr$ which means the actual signal strength.

For occasional packet loss, we use sliding window for packet buffering (see *Samples-Buffer* in Tables 1 and 2). In each broadcast period, if an expected packet is received, the RSSI value is put into the respective sliding window. If the packet is lost, a zero

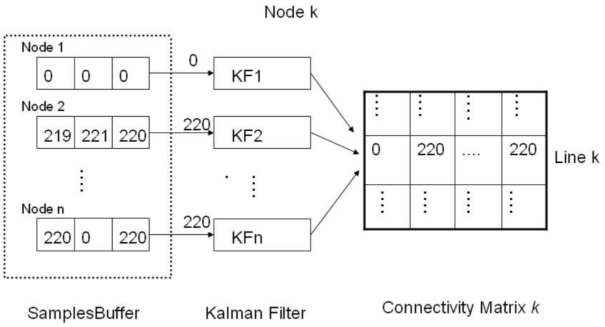


Fig. 5. The filtering process for RSSI

value is inserted into the window. Then, before broadcasting, the non-zero values in the window are averaged and used as the effective RSSI value for that link. When all values in the window are zero, the link is considered broken. The filtering process is illustrated in Figure 6.

5 Experimental Results

We implemented the propagation process of extended connectivity matrix into MICAz motes. The version of TinyOS environment we used is 1.1.15, which is also the last version before TinyOS 2.0. The program running in each MICAz node is identical except the unique node ID. The source code is compiled in Cygwin environment and uploaded into the node via a MIB600 Ethernet programming board. The period of connectivity matrix propagation for each node is set to be one second.

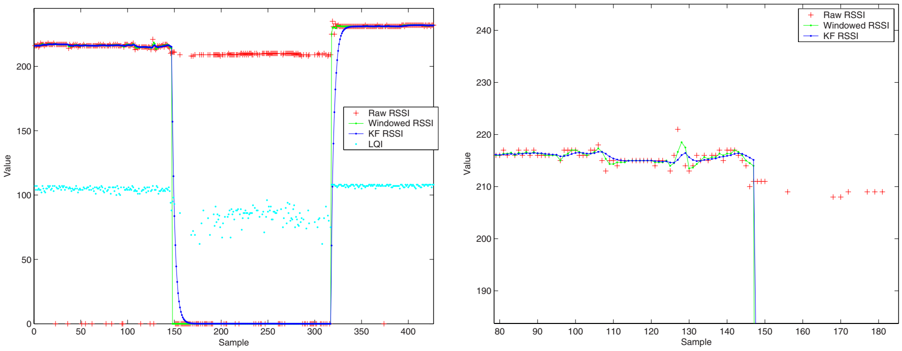
For sensing data retrieval, we connect arbitrary node to the MIB600 board. Then the MIB600 forwards the data received from UART via a TCP/IP interface. A PC program is designed to get the sensing data from programming board via TCP/IP port, and save the data periodically into a local file. Finally, a Matlab program reads the local file periodically, and performs generation of relative positions and adjustment of coordinates system. The reason we don't use Matlab to connect TCP/IP directly is it is very inefficient and slows down the whole computing process.

The RF module on MICAz, namely CC2420, provides flexible transmitting power settings ranging from -25 dBm to 0 dBm. We set the transmitting power of each unit to -15 dBm. Then the RF range of nodes in an indoor lab would be approximately 6-10 meters. We conducted our test in a $5m \times 5m$ area in which the RSSI reading is around 200 to 270. For CC2420, only lower 8 bits of RSSI readings will be presented, etc. 270 will be presented as $0E_{(16)}$. So for very small RSSI readings (smaller than 56), we add an offset of 256 for it. The LQI values, according to our observation, are typically above 100 when two nodes are in good link state, and drop dramatically below 70 when nodes are in bad link state. In our TinyOS program, we discard packets with LQI below 100.

5.1 The Effect of Data Filtering

We place two nodes on the floor, one sender and one receiver, with an initial distance of 2 meters. The sender sends 145 packets to the receiver in the first 145 seconds. Then we move the sender to 5 meters away from the original place, and collect 172 packets in around 3 minutes. After that, the sender is placed in a position 1 meter away from the receiver, and another 110 packets are collected.

Here we use a 3-byte sliding window for RSSI buffering. For every sensing period, the window filter checks the last 3 RSSI readings and provides the average of the positive non-null readings to the Kalman filter. If the buffer contains zeros only, the filter reports zero RSSI value. The Kalman filter takes the values reported by the sliding window filter as the newest measurement, and keeps estimating the RSSI state. The filtering results are presented in Figure 6. In Figure 6(a), we observe the packets handling along three stages with different distances. In the first stage (2 meters), 12 packets were lost out of 145 and only two were consecutive. The reported LQI values in the first stage are



(a) Overall Effect of Filtering Effect

(b) The Effect of Window filter and Kalman filter

Fig. 6. Filtering Result

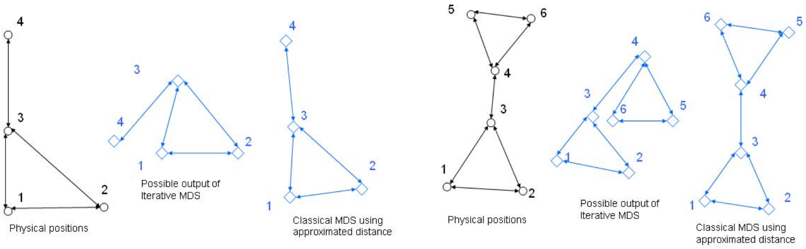
always above 100. The occasional packet missing is successfully filtered by the sliding window. In the second stage (5 meters), 67 of 172 packets were lost during the transmission and only two thirds of RSSI values are measured. All LQI values are below our threshold 100, so no RSSI value is counted as valid measurement. Results of window and Kalman filtering drop to zero in the second stage. In third stage, two nodes are very close to each other (1 meter). We observe nearly perfect packet transmission rate (only one lost) and more stable LQI value compare to previous stages.

Notice that when the RSSI steps up or down, the result of window filtering can timely track the RSSI value, and the result of Kalman filter shows further smoothness. The RSSI values fluctuate even when nodes position are fixed, as shown in Figure 6(b), our window and Kalman filters can generate smoother curves. This property enables that when mobile units are stopped we can generate stable relative positions, and when units are moving we can derive the direction and speed of their mobility.

5.2 Relative Position Under Missing Measurement

Here we study the result of relative positioning under missing measurements. In some cases, a group of mobile units are connected but not fully linked. As shown in Figure 7(a), unit 1,2 and 3 are connected with each other, but unit 4 can only communicate with unit 3. The RSSI measurements between 1 and 4, 2 and 4 are missing in this case. However, the missing measurement implies an *implicit constraint* on the pair of units. That is, if a pair of units can not receive packets from each other, their pairwise distance in signal strength space should be larger than a certain threshold. We compare the performance of iterative MDS and classical MDS under missing measurement.

The iterative algorithm described in [7] is used to generate a set of relative positions. This iteration process try to minimize the stress function described in Equation (3). It keeps iterating until the difference between $\sigma(X^{k-1})$ and $\sigma(X^k)$ is smaller than a empirical threshold ϵ. In our experiment, we set ϵ as 2% of the communication range under minimum transmitting power.


Physical positions
Possible output of Iterative MDS
Classical MDS using approximated distance
1
2
3
4
5
6

(a) relative position for 4 units (b) relative position for 6 units

Fig. 7. Relative positioning under missing measurement

In the previous section, we proposed a method to approximate missing measurements by assigning them the minimum accumulated *Signal Distance*. As long as the group of mobile units is connected, the missing pairwise distances are generated using the collected signal strength information.

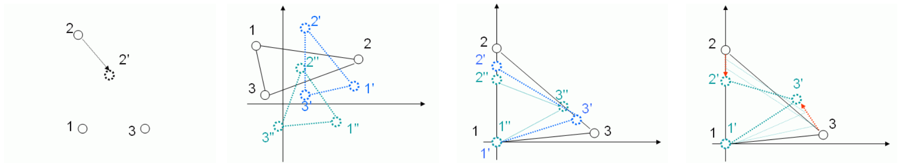
Figure 7(a), center, shows one possible result of iterative MDS. Since the distances between 4 and 1, 4 and 2 are not constrained, unit 4 can be placed in any position as long as the distance between 4 and 3 is compatible with the measurement. However, this result is probably not compatible with the physical placement because in this case unit 4 is so close to units 1 and 2 that they should receive packets from each other. Moreover, incomplete distance information causes ambiguity for positioning determination. For the same set of distance information, each run of iteration generates a different result. This ambiguity makes it hard to capture unit mobility.

Figure 7(a), right, illustrates the result from classical MDS with approximated distances. The relative positions of all units are compatible with the physical positions. Due to the approximation, the estimated distance of 1 and 4, and 2 and 4 are larger than the actual ones. So classical MDS leads unit 4 to a farther position, but the result is still acceptable. Another nice property of our method is that the solution for every set of distance information is unique, which enables us to manage units mobility.

Figure 7(b) provides another example with more loosely connected units. Only 7 RSSI readings are available among 15 possible pairs. Figure 7(b), center, shows one possible result generated by iterative MDS. The lack of enough constraints leads to even more ambiguity of positions. Units 1 and 2 are placed too close to unit 5 and 6, which makes it an unacceptable solution. On the other hand, using the approximated distance information, the classical MDS generates a set of reasonable relative positions as shown in Figure 7(b), right. Again, we observe that the approximated distances between units are larger than the actual ones. But the relative position is still acceptable when 50% of the RSSI information is missing.

5.3 Relative Position Under Mobility

Using our proposed approximated distance, classical MDS can generate unique solution for a set of RSSI readings. Here we conduct experiments on generating relative



(a) (b) (c) (d)

Fig. 8. Smoothing the moving trajectory

positions when units are moving, and also managing their moving pattern in signal strength space. For simplicity, only one unit is allowed to move.

Smoothing the trajectory. Figure 8(a) depicts the physical position of unit 1, 2 and 3. Unit 2 moves slowly to the dashed position 2′. Figure 8(b) shows the dynamic relative position results without Kalman filtering and coordinates adjustment. The black circles denote the initial position of the units, and blue dashed circles denote a set of typical intermediate positions. Final positions are illustrated by green dashed circles. But the calculated positions are not stable both in initial and final stage. The fluctuating RSSI readings makes the matrix calculation of MDS generate a totally different result.

Figure 8(c) aligns the positions according to the rule of coordinates adjustment. Jumping still exists but generally the positions follow the trajectory of $(1, 2, 3) \rightarrow (1', 2', 3') \rightarrow (1'', 2'', 3'')$.

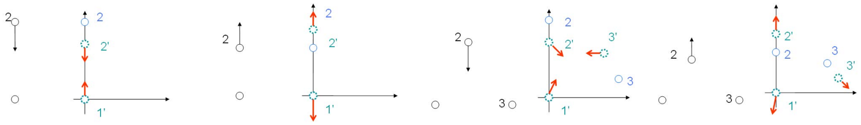
Figure 8(d) shows the results of Adding kalman filter to the RSSI readings. We obtain rather nice moving trajectory of the units. The red dashed lines indicate the moving pattern of unit 2 and 3. Jumping of unit positions is eliminated since the RSSI readings of each pair of units are changing smoothly. And a set of smoothly changing distances can generate a set of smoothly changing relative positions.

Managing the mobility. In this case we study the difference of consecutive relative positions in order to obtain the moving pattern of mobile units, which is particularly useful for autonomous mobile units, e.g., to drive their movement one to another. For each new estimate of relative positions, we compare the pairwise distance of units in signal strength space to the previous one and calculate a vector denoting the moving direction for each unit i:

$$\mathbf{V}_i = \sum_{j \in \mathbb{N}, j \neq i} (d_{ij} - d_{ij}^{-}) \times (\mathbf{X}_i - \mathbf{X}_j) \qquad (8)$$

Figures 9(a) and 9(b) show the case of two mobile units. Black circle denotes physical position and arrow denotes physical moving direction. In signal strength space, we have the green dashed circles for the current position and red arrows for direction vector. Note that although unit 2 is not moving physically, the obtained velocities indicate that both units are approaching each other or moving apart, which is correct given the relative nature of the determined positions.

Figures 9(c) and 9(d) depict another example with three mobile units. When unit 2 moves closer to, or away from, units 1 and 3, the direction vectors of all three units correclty indicate a convergence or divergence among them.



(a) (b) (c) (d)

Fig. 9. Managing relative mobility

The moving vectors **V** are probably the main result of this paper, because they allow establishing the connection between the relative positions space and the physical space. When unit **a** wants to move closer to unit **b** or group **C** of several units, it can perform a tentative physical move and obtain feedback from the corresponding moving vector $\mathbf{V}_a$ to adjust the moving direction.

6 Conclusions

In this paper, we first address challenges and difficulties of managing relative positions for *mobile* units. Then we explore the idea of connectivity tracking and propose an extension for Zigbee platform. Multidimensional scaling method is used for computing relative positions. We propose an approximation method to generate pairwise distance under missing measurement. Comparing with iterative MDS method, our method can generate more precise relative position results, with little computation cost. We also combine sliding window and Kalman filter to smooth the noisy signal strength readings. Filter techniques together with coordinates adjustment, generates smooth moving trajectory which can indicate the physical movement and support coordinated actions. We are currently connecting the MICAz nodes to real autonomous mobile robots enabling them with coordinated movements capabilities.

Acknowledgement

This work is supported by China-Portugal Cooperation Project: Managing Network QoS in Distributed Computer Control Applications, the National Natural Science Foundation of China under Grant No.60434030, the National High-Tech Research and Development Plan of China under Grant No.2006AA01Z218 and National Research Foundation for the Doctoral Program of Higher Education of China under Grant No.20050335020.

References

1. Desai, J., Ostrowski, J., Kumar, V.: Controlling Formations of Multiple Mobile Robots. In: Proceedings of the IEEE International Conference on Robotics & Automation (1998)
2. Coutinho, F., Barreiros, J., Fonseca, J.: Choosing Paths that Prevent Network Partitioning in Mobile Ad-hoc Networks. In: 5th IEEE International Workshop on Factory Communication Systems (2004)

3. Wang, G., Cao, G., Porta, T., Zhang, W.: Sensor Relocation in Mobile Sensor Networks. In: Proceedings of the IEEE INFOCOM (2005)
4. Savvides, A., Han, C.C., Srivastava, M.B.: Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors. In: Proceedings of MOBICOM 2001 (2001)
5. Priyantha, N.B., Chakraborty, A., Balakrishnan, H.: The Cricket Location-Support System. In: Proceedings of MOBICOM 2000 (2000)
6. Niculescu, D., Nath, B.: Ad Hoc Positioning System (APS) using AOA. In: Proceedings of the IEEE INFOCOM (2003)
7. Ji, X., Zha, H.: Sensor Positioning in Wireless Ad-hoc Sensor Networks Using Multidimensional Scaling. In: Proceedings of the IEEE INFOCOM (2004)
8. Patwari, N., Hero, A.O., Perkins, M., Correal, N., O'Dea, R.J.: Relative Location Estimation in Wireless Sensor Networks. IEEE Trans. on Sig. Proc.Special Issue on Signal Processing in Networking 51, 2137–2148 (2003)
9. Shang, Y., Ruml, W., Zhang, Y.: Localization From Mere Connectivity. In: Proceedings of the ACM MobiHoc (2001)
10. Shang, Y., Zhang, Y.: Improved MDS-Based Localization. In: Proceedings of the IEEE INFOCOM (2004)
11. Shen, X., Wang, Z., Sun, Y.: Connectivity and RSSI Based Localization Scheme for Wireless Sensor Networks. In: Huang, D.-S., Zhang, X.-P., Huang, G.-B. (eds.) ICIC 2005. LNCS, vol. 3644, Springer, Heidelberg (2005)
12. Hu, L., Evans, D.: Localization for Mobile Sensor Networks. In: Proceedings of the ACM MOBICOM (2004)
13. Crossbow: Micaz, http://www.xbow.com/Products/productdetails.aspx?sid=164
14. Srinivasan, K., Levis, P.: RSSI Is Under Appreciated. In: Proceedings of the EmNets (2006)
15. Chipcon: CC2420 Manual, http://www.chipcon.com/files/CC2420_Brochure.pdf
16. Facchinetti, T., Buttazzo, G., Almeida, L.: Dynamic Resource Reservation and Connectivity Tracking to Support Real-Time Communication among Mobile Units. EURASIP Journal on Wireless Communications and Networking 5, 712–730 (2005)
17. Welch, G., Bishop, G.: An Introduction to the Kalman Filter. In: ACM SIGGRAPH Course Notes (2001)

The Implementation of a Fully Integrated Scheme of self-Configuration and self-Organization (FISCO) on Imote2

Jialu Fan[1], Jiming Chen[1], Jialiang Lu[2], Yu Zhang[3,⋆], and Youxian Sun[1]

[1] State Key Laboratory of Industry Control Technology,
College of Info Science and Engineering,
Zhejiang Univ., Hangzhou 310027, P.R. China
Tel:+86 571 8795 3762
yu.zhang@orange-ftgroup.com
[2] ARES INRIA / CITI, INSA-Lyon, F-69621, France
[3] France Telecom R&D Beijing

Abstract. Wireless Sensor networks are gaining a role of importance in the research community. In this paper, we choose Imote2 to establish our testbed according to the analysis and the comparison of the hardware capabilities and software characteristics of the current WSNs products. Moreover, we present our hardware, software platforms and "30 motes" testbed to validate and evaluate our Fully Integrated Scheme of self-Configuration and self-Organization (FISCO) while further propose the rudiment of our general routing and data dissemination testbed. In the test stage, experimental results show that each mote in the network obtains the corresponding role and takes correct actions to handle every receiving message.

1 Introduction

As wireless sensor networks have emerged as an exciting new area of research in computer science, many of the logistical challenges facing those who wish to develop, deploy, and debug applications on realistic large-scale sensor networks have gone unmet. Deploying a network into a realistic environment requires iteratively reprogramming dozens of nodes, locating them throughout an area large enough to produce an interesting radio topology, and instrument them to extract debugging and performance data. Although reasonable tools exist for evaluating large sensor networks in simulation[1], [2], [3], only a real sensor network testbed can provide the realism exigent to understand resource limitations, communication loss, and energy constraints at scale. Today, practically every research group and every larger sensor network project is using a platform for implementation work spanning a great variety of approaches and hardware architectures. This is primarily due to the large application space with differing requirements on resources and architectures each on the one hand and the goal to minimize the overhead on the other hand.

⋆ Corresponding author.

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 672–682, 2007.

There are many platforms for WSNs, which are both commercial solutions and research projects. We find some successful platforms of WSNs, such as the Intel Mote2(Imote2), Mica2, Atific Helicopter, Tmote Sky and μAMPS. Mica2 is one of the most popular and commercially available sensors which is marketed by Crossbow Technology[4]. The Mica2 Motes use an Atmega128L micro-controller and the Chipcon CC1000, FSK modulated radio. Atific Helicopter is the world's first multi-radio platform for WSNs, which is marketed by Atific Corporation[5]. It integrates Altera Cyclone EP1C20 FPGA and Nordic Semiconductor nRF2401A radio. Tmote Sky, which is marketed by Moteiv Corporation[6], uses MSP430 F1611 microcontroller and Chipcon CC2420 radio. The μAMPS (micro-Adaptive Multi-domain Power-aware Sensors) node is a non-commercial wireless sensor device developed by MIT[7]. The μAMPS node is based on the StrongARM SA-1110 microprocessor and its own radio. Unfortunately, the above four platforms support TinyOS software development only. However, the Imote2 is the newest one among these sensor nodes. It is developed and marketed by Intel Corporation and Crossbow Technology[8]. The Imote2 platform is built around a PXA271 XScale processor. It integrates ChipCon 2420 onto the main board and a built in 2.4 GHz antenna. Furthermore, TinyOS, Linux and SOS are all able to run on Imote2. According to the analysis and the comparison of the hardware capabilities and software characteristics of these platforms, we choose Imote2 to establish our testbed for its powerful functionality.

WSNs are spontaneous networks formed by sensor nodes with limited information processing capabilities and very constraint energy resources. Communications are enabled in an ad hoc fashion using low-power wireless communication technologies such as IEEE802.15.4[9]. In WSNs, there is neither pre-defined communication structure nor centralized control element. In order to set up a connected efficient network without any human intervention, two mechanisms are essential: self-configuration and self-organization. The first one deals with dynamic distributed address allocation and configuration of network parameters[10],[11] whereas the second one aims to structure the network to provide efficient communication[10],[12],[13]. Hence, we have proposed in a former paper[14] the first fully integrated scheme of self-configuration and self-organization (FISCO). It can better resolve the distributed address allocation using the structure generated by itself with low message overhead. It also provides significant energy savings comparing to other schemes and low latency. Due to the superiority of immediate configuration, low message cost and low energy consumption and long lifespan on FISCO, we firstly implement the FISCO on Imote2 testbed. Moreover, through testing FISCO, we improve the system service on testbed.

The paper is organized as follows. Section 2 describes the join procedure of the Fully Integrated Scheme of self-Configuration and self-Organization (FISCO) since we implement this procedure in the current stage. We illustrate our implementation on Imote2 in Section 3, which contains the hardware, system service and the testbed. In Section 4, we bring forth the experimental results. Finally, we give the future work on the testbed and concluding remarks in Section 5.

2 FISCO

In FISCO, Nodes are classified to three roles: *leader*, *gateway* and *member*. A *leader* is in charge of all communication as well as address allocation in its 1-hop neighborhood. *Leaders* are not directly connected in the network. *Gateways* interconnect *leaders* which are separated with 2-hop distance. *Leaders* and *gateways* form a virtual backbone in the network. *Members* are 1-hop neighbors of *leaders*. Only *leaders* send periodical messages to their 1-hop neighbors, while no periodic control traffic is created by other nodes.

A two-level hierarchical address allocation scheme is used to guarantee the uniqueness of assigned addresses. On the lower level, a *leader* assigns an address to its newly arrived neighbor node using a stateful function $f(n)$. On the higher level, the address space in the network is distributed among all *leaders*. A full address space is divided into address pools. Each *leader* owns at least one address pool. The design of FISCO combines both high level and low level address allocation to ensure the uniqueness of address in a partition. Address allocation also takes advantage of existing self-organization structure (*leader*-*gateway* backbone). If a new node arrives in the 1-hop neighborhood of a *leader* which has no more address available in its address pool, then it will allocate the next address pool which is known by all *leaders* in the partition for itself and send update information to other *leaders* through the *leader*-*gateway* backbone.

FISCO is an event-driven scheme including joining, departure and partition management procedures. Local re-organization is also proposed, which allows *leaders* and *gateways* switch to *members* for energy saving. In the following, we elaborately illustrate the joining procedure as well as *leader*, *gateway* and *member* handling messages in this procedure because our implementation spans these parts in the current stage. A newly arrived node will take at most the following three stages to be integrated into the network organization.

Joining of A Member: The new node waits a timeout for a *Leader-Broadcast-Msg*. By detecting this message, it performs a 1-hop address allocation request directly with the *leader* using unicast messages. The *leader* uses its stateful function to assign an address from its address pool to the new node and requires an *Address-Ack*. The new node becomes a *member*.

Joining of A Leader: If no *leader* is detected, the new node sends a 1-hop broadcast *Member-Solicitation-Msg*. It is used to look for a configured node (*member* or *gateway*) in its radio vicinity. Upon receiving a *Member-Advertisement-Msg* from a neighbor, it enters into a 2-hop address allocation procedure with the *leader*. The *leader* will allocate a new address pool from the address space (shared by all *leaders*) to the new node. At the same time, it propagates the update information using virtual backbone to other *leaders* in the partition. The new node becomes a *leader* and the intermediate node changes itself to a *gateway* in order to connect the two *leaders* in the virtual backbone. This scenario can also be considered as an expansion of the FISCO structure.

Partition Setup: If no neighbor is detected, the new node assigns itself as the *first leader* with an address pool and an address. It generates a random partition identifier (PartitionID). Hence a new partition appears in the service area.

The design of joining procedure in FISCO shows the fully use of existing organization for newly arrived nodes. FISCO lets each new node firstly discover the network structure within its neighborhood. Then it achieves address allocation and integrates itself into the structure by using only unicast messages.

3 Implementation

Our implementation spans hardware and software to explore and demonstrate FISCO via WSNs. In the hardware section, we introduce our current embedded network devices, Intel Mote2 (Imote2). Then, in the software section, we discuss the universal operating system (OS), Linux, and system service architecture. Finally, we survey our current and future testbed for interacting and learning at the whole-system level.

3.1 Hardware

Imote2 is an advanced wireless sensor node platform. The platform is built around an XScale processor, PXA27x. It integrates an 802.15.4 radio onto the main board (ChipCon 2420) and a built in 2.4 GHz antenna. It exposes a “basic sensor board” interface, consisting of two connectors on one side of the main board, and an “advanced sensor board” interface, consisting of two high density connectors on the other side of the main board[8]. The Imote2 is a modular stackable platform. The main board (containing the processor and the 802.15.4 radio) can be stacked with sensor boards to customize the system to a specific application, along with a “power board” to supply power to the system. All together, the hardware platforms have been sufficient to meet our needs of both research and experimentation.

3.2 System Service

Currently, the Imote2 is supported by several operating systems including TinyOS 1.1, TinyOS 2.0, SOS and Linux. Due to the universalitywe choose Linux operation system and build our embedded software with standard C language. The basic structure of the system is shown in Fig. 1.

The physical layer is represented by the hardware, Imote2. The MAC layer is implemented in the Linux driver level. We use fixed ID number saved in file system as the MAC address, which is set by *Init* module in network layer (as shown in Fig. 2). When receiving the same packets (check destination address and source address), the MAC layer will only transmit the first packet to network layer but ignore the remainder. The network layer is implemented in the Linux

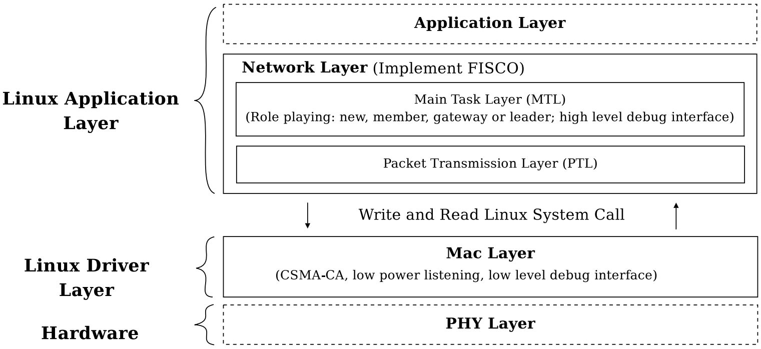

Linux Application Layer
Linux Driver Layer
Hardware
Application Layer
Network Layer (Implement FISCO)
Main Task Layer (MTL)
(Role playing: new, member, gateway or leader; high level debug interface)
Packet Transmission Layer (PTL)
Write and Read Linux System Call
Mac Layer
(CSMA-CA, low power listening, low level debug interface)
PHY Layer

Fig. 1. The Basic Structure of The System

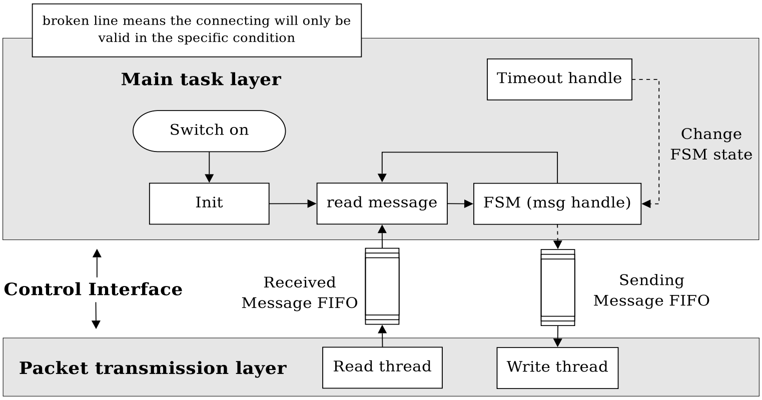


Fig. 2. Module Description in Network Layer

application layer. Due to the lack of sensor on Imote2, issues of sensing application are on the agenda but are currently unaddressed in the architecture. In the future, we will add the application layer upon the network layer in the Linux application layer (see Fig. 1). In other words, we will build the protocol stack in the Linux application layer. The network layer contains Main Task Layer (MTL) and Packet Transmission Layer (PTL). The functionality in the PTL is that the receiving node can check if its address matches that of the packet, and it will transmit the packets for it to the MTL, and discarding the packets not for it. The routing protocol is implemented in the MTL. Fig. 2 further shows the relationship between the modules in the network layer. The MTL is message-driven scheme.

The *Init* module is in charge of initiating hardware (using open system call) and system status in network layer. It creates packet transmission threads. After initiating succeeds, the execution enters into *read message* module. The responsibility of *read message* module is blocking read from *received message FIFO*. After

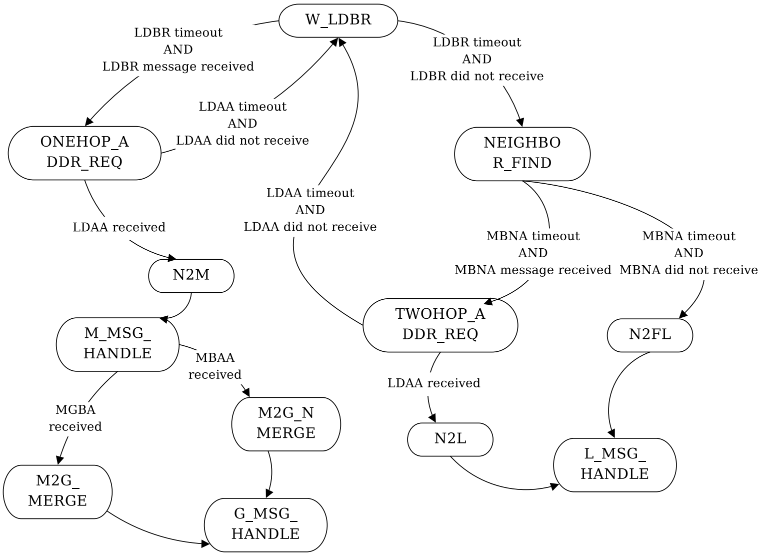


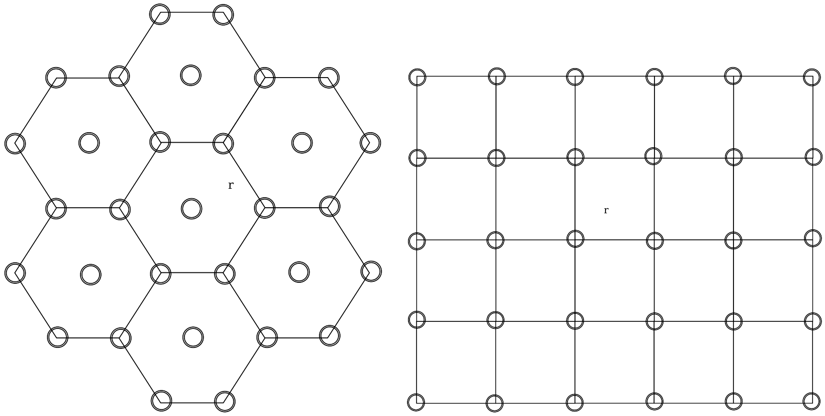
Fig. 3. The Details of The Finite State Machine(FSM) Module

reading the message, the execution enters into *FSM* (Finite State Machine) module. Fig. 3 shows the details of the *FSM* module. *FSM* module handles messages passed by *read message* module and send message to *sending message FIFO* in case of necessity. Upon receiving different messages, *FSM* will take corresponding actions. After the message handled, MTL returns to *read message* module. Furthermore, we set a *Timeout handle* module to handle timeout event which maybe changes *FSM* state. This module must run after the current message handled in *FSM* in order to avoid collision between message handle event and timeout event. The timer is set by *FSM* or by *Timeout handle* module.

3.3 Testbed

Our current experimental platform is functional but limited when compared to the application scope of FISCO. We implement the join procedure, *leader*, *gateway* and *member* handling messages on the testbed. It is the result of a focused effort to produce a solution for a set of particular goals rather than to provide a general framework. To that end, it exists more as a proof that a highly constrained FISCO is achievable and that Linux and standard C language provide a suitable platform for development.

Our testbed has been deployed on a network platform of 30 Imote2 nodes. In order to facilitate the collection of experimental result, we deploy two kinds of setups for the testbed as shown in Fig. 4. Fig. 4(a) is a hexagon while Fig. 4(b) a uniform grid. In the scenario of Fig. 4(a), we switch on the nodes from center to margin in order to observe the actions they take. However, we switch on the nodes from one side to the other side in scenario Fig. 4(b) to estimate that.



(a) Hexagon Platform (b) Uniform Grid Platform

Fig. 4. Setup for The "30 motes" Testbed Platform

4 Experimental Result

As mentioned before, we implement the join procedure of FISCO on our testbed. However, in order to test the join procedure, we should build the mature FISCO network, in other words, the current network sometimes needs to contain *leaders*, *gateways* and *members* when new node arriving. Hence, we implement the functionality of handling messages on *leader*, *gateway* and *member* as well.

New node configuration generally consists of the following three procedures. A new node will first listen to the medium during an interval of time, denoted as *LDBR-TIMOUT*. If it detects *Leader-Broadcast-Msg*(LDBR), then it will enter in the **one-hop address allocation**. Moreover, if it does not detect any LDBR, then it considers that there is no *leader* in its neighborhood. Hence it will try to find any *Member/Gateway* nodes. It sends a *Member-Notification-Msg*(MBNS) and waits for *Member-Advertisement-Msg*(MBNA) during a *MBNA-TIMEOUT*. If it detects MBNA, then it will enter in **two-hop address allocation**. If still no MBNA detected, it will decide to **create a partition** in the network and assign itself the first address and the first address pool in the address space. It also generates a random PartitionID for its partition and becomes a *first leader*.

In the implementation procedure, the failed adderss allocation will result in the waste of address space, so we set *one-hop-buffer* and *two-hop-buffer* to record the allocated address on *leaders*. The failed allocated address(without Ack) will be reused in the next address allocation procedure. In the testing procedure, we test the following eight scenarios:

□ First node configuration

□ Configure to member

□ Fail to configure to member

□ Reuse of no-acked address in one-hop buffer

□ Two-hop address configuration with the only leader in the partition

□ Fail to configure to leader node

□ Reuse of on-acked address in two-hop address configuration

□ Two-hop address configuration with a backbone (multiple leaders)

Through testing above eight scenarios, the FISCO implementation shows its correctness. We then test the whole join procedure,in other words, the new node transfers to *member*, *leader* and *first leader*. In the following, we illustrate the One-Hop Address Allocation, Two-Hop Address Allocation and Partition Setup.

4.1 One-Hop Address Allocation

The new node enters into the network and receives the LDBR(s) in the *W-LDBR* state. Hence it sends *Member-Address-Request-Msg*(MBAR) to the *leader*(generally the one with smallest address). The selected *leader* allocates an address from its address pool as receiving the MBAR, and then sends the *Leader-Address-Allocation-Msg* (LDAA) to the new node. The new node changes itself to a *member* and updates its address according to the receiving LDAA.

A Example of Processes of A New node to Member

```
FISCO begin...
[State] W_LDBR
        Collects x LDBR(s)
[State] ONEHOP_ADDR_REQ
        Sends MBAR to one of the x leader(s)
        Waits for the LDAA from that leader...
        Receives the corresponding LDAA
[State] N2M
[State] M_MSG_HANDLE
```

4.2 Two-Hop Address Allocation

The new node enters into the network and doesn't receive the LDBR in the *W-LDBR* state but receive the MBNA in the *NEIGHBOR-FIND* state. It sends MBAR to the neighbor (*member*/*gateway*). If the new node receives MBNAs from member as well as gateway, it choose gateway to reduce the number of dominating nodes. The neighbor who receives the two-hop MBAR forwards the MBAR to its *leader*. The *leader* allocates an address pool to the new node through the forwarding node and sends the *Leader-Address-Update-Msg*(LDAU) to the other *leaders* in the partition through the virtual backbone.

A Example of Processes of A New node to Leader

```
FISCO begin...
[State] W_LDBR
        Collects zero LDBR
[State] NEIGHBOR_FIND
        Receives the MBNA(s)
[State] TWOHOP_ADDR_REQ
        Sends MBAR to one neighbor
        Waits for the LDAA from that leader...
        (The neighbor forwards the MBAR to its leader)
        (The leader allocates one address pool to the new node)
        (The leader sends LDAUs to other leaders)
        Receives the corresponding LDAA
[State] N2L
[State] L_MSG_HANDLE
```

4.3 Partition Setup

The new node enters into the network. It doesn't receive the LDBR in the *W-LDBR* state and doesn't receive the MBNA in the *NEIGHBOR-FIND* state. It then changes its state from new to the *first leader* of the partition and handles *leader*'s messages. It sends LDBR periodically. If the *first leader* receives a 1-hop MBAR, it allocates an address from its current address pool to the new node. Here, we use the stateful function of address allocation $f(n) = f(n-1)+1$. The *first leader* sends one-hop *Leader-Address-Allocation-Msg*(LDAA). Furthermore, the new address allocated to the new node is 1 bigger than the last allocated address.For example, if the last allocated address is $(0x10000)$ the new allocated address is $(0x10001)$.

A Example of Processes of A New node to First Leader

```
FISCO begin...
[State] W_LDBR
        Collects zero LDBR
[State] NEIGHBOR_FIND
        Collects 0 MBNA
[State] N2FL
[State] L_MSG_HANDLE
        Sends LDBR periodically
        Receives one MBAR
        Sends one address from its address pool back
[State] L_MSG_HANDLE
```

5 Conclusion and Discussion

Based on the comparison and analysis, we choose Imote2 platform to establish our testbed. We then describe our hardware and software platforms and the

"30 motes" testbed. We implement the join procedure and the functionality of handling messages on *leader*, *gateway* and *member*.

What we would like to do is to use the WSNs hardware and software architecture to implement the application of FISCO and other routing protocols in a more versatile, general framework. Beyond these scenarios, we look forward to expanding our understanding of whole-system behavior through formalism and parameterization of distributed WSNs.

In the future, we will focus on implementation of the remainder of FISCO on our testbed and add the application layer in the protocol stack to test the data dissemination in the network. Moreover, we will enrich the testbed to satisfy more distributed routing protocols in WSNs.

Acknowledgements. The authors thank the anonymous reviews, whose comments improve the quality of the paper, and the financial support of France Telecom R&D, Natural Science Foundation of China under Grant No.60604029,

References

1. Breslau, L., Estrin, D., Fall, K., Floyd, S., Heidemann, J., Helmy, A., Huang, P., McCanne, S., Varadhan, K., Xu, Y., Yu, H.: Advances in Network Simulation. IEEE Computer 33(5), 59–67 (2000)
2. Riley, G.F., Ammar, M.H., Fujimoto, R.: Stateless Routing in Network Simulations. In: Proc. of the International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, IEEE, San Francisco, CA, USA (2000)
3. Mallanda: SensorSimulator: A Simulation Framework for Sensor Networks. Thesis of Master, Louisiana State University (2005)
4. Crossbow Corp. home page, Mica2-Wireless Measurement System, Available at: http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICA2_Datasheet.pdf
5. Atific Corp. Atific Helicopter High Performance Multi-Radio WSNs Platform, Available at: http://atific.fi/files/helicopter/AtificHelicopter_WhitePaper.pdf
6. Moteiv Corp. home page, Low Power Wireless Sensor Module, Available at: http://www.moteiv.com/products/docs/tmote-sky-datasheet.pdf
7. Calhoun, B.H., Daly, D.C., Verma, N., Finchelstein, D., Wentzloff, D.D., Wang, A., Cho, S.-H., Chandrakasan, A.P.: Design Considerations for Ultra-low Energy Wireless Microsensor Nodes. IEEE Transactions on Computers, 727–749 (2005)
8. Crossbow Corp. home page, Imote2 - High Performance Wireless Sensor Network Node, Available at: http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/Imote2_Datasheet.pdf
9. LAN MAN Standards Committee of the IEEE Computer Society: IEEE Std802.15.4 Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low-Rate Wireless Personal Area Network (LR-WPANs). IEEE-SA Standards Board, (2003)
10. Adjih, C., Jacquet, P., Viennot, L.: Computing Connected Dominated Sets with Multipoint Relays. Ad Hoc and Sensor Wireless Networks 1, 27–39 (2005)

11. Mohsin, M., Parkash, R.: IP address assignment in a mobile ad hoc network. In: IEEE Military Communications Conference (MILCOM), Anaheim, USA (October 2002)
12. Wu, J., Li, H.: On calculating Connected Dominating Set for efficient routing in ad hoc wireless networks. In: Proc. Of the Third Int' Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, Seattle, USA, pp. 7–14 (August 1999)
13. Younis, O., Fahmy, S.: HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. IEEE Transactions on Mobile Computing 3(4), 366–379 (2004)
14. Lu, J., Valois, F., Bartheland, D., Dohler, M.: FISCO: a Fully Integrated Scheme of self-Configuration and self-Organization for WSN. In: IEEE Wireless Communications and Networking Conference (WCNC), Hong Kong (March 2007)

Truthful Resource Allocation in Selfish Sensor Web

Yong-Kang Ji, Yi Zhang, Zhicheng Xu, and Min-You Wu

Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, 200030, P.R. China
{jyk, nightmarex, xuzhicheng, mwu}@sjtu.edu.cn

Abstract. Sensor Web, as an extension of Sensor Networks, is expected to be used in the distributed, large-scale environment. With the help of Internet, its application can even be extended to the world wide. And in the distributed environment, a Sensor Web usually consists of different sensor networks or single sensor owned by different organizations such as research institutions, companies or even individual user. So maximizing their benefits is what the sensors most concern about. In other words, the sensors in Sensor web are rational and selfish. In this paper, we discuss the selfish problems of Sensor Web and resolve them using specific designed mechanism. We describe several scenarios of the applications in Sensor Web and consider the variable cost and value issues which are, to the best of our knowledge, first time studied in the Sensor Web. In the end, our mechanisms are proved to be truthful and can be extended to many other applications.

1 Introduction

Sensor Web is a newly developed research field with the development of sensor networks. It demonstrates an amorphous network of spatially distributed sensor platforms. Many corresponding projects have been focus on this field, such as the “JPL Sensor Webs project” [1], “Microsoft SenseWeb Project” [2] and so on. The “JPL Sensor Webs project” tried to build a sensor platform for environment monitoring and some other kinds of applications. And the purpose of “Microsoft SenseWeb Project” is to “browse the physical world in real time” [2]. What’s more, Microsoft built up an online portal: SensorMap [3] where people can easily publish their data getting from the sensor and share with others.

Although these projects focus on different field, they all assume that the sensor of each terminal node would like to coordinate with other sensors, such as sharing data, interacting and so on. However, Sensor Web is a widespread distributed system and maybe consists of many sensors belonging to individual user who does not know with other users at all. They would like to do the things that benefit themselves most, which means they are rational and selfish. Apparently, sharing data or interacting with other sensors for free will only consumes the own resources of one sensor and benefits other sensors, which is not acceptable by a rational and selfish user who owns the sensor. Therefore, Sensor Web needs to provide incentives to encourage every user to contribute their resources to other users, otherwise the Sensor Web cannot be

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 683–693, 2007.

maintained and will be broken down. But the question is: how to calculate the incentives? We need every user to give their true cost and then calculate the corresponding payment. And how can we make sure that every node gives its cost truthfully? We utilize the Game Theory of economics and design a mechanism to guarantee this point. In this paper, we discuss the selfish problems along with several classical scenarios of Sensor Web, and then design truthful mechanisms to resolve the selfish problems. What's more, we will take the variable cost and value into account, which is the first time discussed in the Sensor Web.

The rest of the paper is organized as follows. The selfish motivation in Sensor Web and theory of mechanism are addressed in Section Ⅱ. Then we introduce our mechanism and prove them truthful in Section Ⅲ. Some simulation results will be demonstrated and analyzed in Section Ⅳ. Finally, Section Ⅴcloses this paper with some brief concluding remarks.

2 Preliminary

2.1 The Selfish Motivation in Sensor Web

The selfish problems of traditional s*ensor networks* have been studied before [4]. But the sensors in a sensor networks are usually obedient because they all belong to the same organization. E.g., the "smart dust" applied in the battle might owns thousands of sensors, and all these sensors obvious belongs to one side of the war. So it is not reasonable to state the selfish of a single sensor in these scenarios. However, different sensor networks may own by different organizations, so it is quite reasonable to address the selfish problems among several sensor networks, and Wu [5] has done some works on this problem.

When refers to the Sensor Web, the selfish problem is quite natural to address. In order to build a widespread system, several Sensor Web projects e.g., [3] utilize Internet as their infrastructure platform which is easy to use and extensible. Like in [3], user who own a camera connected to the internet can easily join the Sensor web by providing the URL of the camera. Then other users in the Sensor Web can get the video catched by the camera. The problem is why user would like to share his camera with others. After all, user has to pay the internet fee, electricity fee and sharing the sensor might occupy computer resources and internet bandwidth. We admit that some users with sharing spirit would like to share his sensor for free; however, most rational and selfish users would like to get payback of sharing. I.e., users have to pay for utilizing the sensor.

What's more, some kind of sensors can be controlled remotely, so people can even publish the control interface which makes other people can also control and adjust the sensor. E.g., people who own a camera can also publish the interface to control the camera, like rotation, zooming in, zooming out and so on. In this case, only one user can control the camera at a time. When several requests of control arrive at one camera at the same time, the competition will happen. In this case, the owner may want to auction the control right in order to maximize his benefits.

In the following part, in order to make our discussion more clear, we will take the sensor of camera as an example. And in this paper, we analysis the ratio of sensor and user, and classify it into two representative scenarios: one sensor, several users and one user, several sensors.

In the first scenario of one sensor and several users, there is only one sensor available and several users want to utilize this sensor. Furthermore, we assume the sensor is exclusive, which means only one user can control the sensor at a time. In order to get the sensor, every user will give a price to show how much it would like to pay for utilizing the sensor. And the sensor will consider each user's bid and the cost of itself and then make a decision. We design truthful mechanisms to make sure that every user will give their truthful price and at the same time the sensor can also maximize its benefits.

In the second scenario of one user and several sensors, there are several sensors that would like to supply service and only one user need the service. Furthermore, we assume the user only need to utilize one sensor at a time. In order to sell its service to the user and get benefits, every sensor will give a price to show how much it would like to charge for the service. And the user will consider the each sensor's bid and value of the service and then make a decision. We design a truthful mechanism to make sure that every sensor will give their truthful price and at the same time the user can also maximize its benefits.

2.2 Mechanism Design Approach

In recent years, multi-agent systems have been extensively studied in both computer science and economics, but the two communities have different approaches [6], [7]. In computer science, agents are typically assumed either to be obedient or to be adversaries. An agent is obedient if it follows the prescribed algorithm, such as a routing algorithm. An agent is adversaries if it plays against each other, deviates from the protocol in arbitrary ways that harm other agents, such as in the mutual exclusion problem. On the other hand, the strategic agents in game theory are neither obedient nor adversarial; they are rational and will respond to well-defined incentives and will not deviate from the protocol unless it improves their gain. With the Mechanism Design approach, each agent in the network determines whether it will execute the task so it will be beneficial. At the same time, the global optimal goal also can be achieved.

When designing and analyzing scenarios in which the participants act according to their own self-interests, a general economic model is as follows [9]. Assume that there are n agents, which could be the distributed computers in a grid network, the network links in a network, or the bidders in an auction. Each agent has some private information t_i known by itself only, called its type. And the type could be its computation capability; could be the price it is willing to pay for a good in an auction. Then the set of n agents define a type vector $t = (t_1, t_2, ..., t_n)$. What agent i reveals is its strategy a_i , and the mechanism computes an output $o = o(a_1, a_2, ..., a_n)$ and a payment vector $p = (p_1, p_2, ..., p_n)$, where $p_i = p_i(a_1, a_2, ..., a_n)$. Here the payment p_i is the "money" (it could be either real or virtual money) given to the participating agent i and it depends on the strategies used by all agents. If $p_i < 0$, it means that the agent has to pay $-p_i$ to participate in the action.

Each agent *i*'s preference is given by a valuation function v_i that assigns a real monetary number $v_i(t_i, o)$ to output *o*. Everything in the scenario is public knowledge except the type t_i, which is a private information to agent *i*. Let $u_i(t_i, o(a), p_i(a))$ denote the utility of agent *i* at the outcome of the game, given its preferences t_i and strategies profile $a = (a_1, a_2, ..., a_n)$ selected by agents. Let $a_{-i} = (a_1, ..., a_{i-1}, a_{i+1}, ..., a_n)$ denote the vector of strategies of all other agents except *i*. A strategy a_i is called dominant strategy if it maximizes the utility for all possible strategies of all agents, i.e.,

$$u_i(t_i, o(a_i, b_{-i}), p_i(a_i, b_{-i})) \geq u_i(t_i, o(a_i^{'}, b_{-i}), p_i(a_i^{'}, b_{-i}))$$

for all $a_i^{'} \neq a_i$ and all strategies b_{-i} of agents other than *i*.

A strategy vector *a* is called Nash Equilibrium if it maximizes the utility when the strategies of all agents are fixed, i.e.,

$$u_i(t_i, o(a_i, a_{-i}), p_i(a_i, a_{-i})) \geq u_i(t_i, o(a_i^{'}, a_{-i}), p_i(a_i^{'}, a_{-i}))$$

for all *i*, and $a_i^{'} \neq a_i$.

A very common assumption in mechanism design is that agents are rational and have quasi-linear utility functions. The utility function is quasi-linear if $u_i(t_i, o) = v_i(t_i, o) + p_i$. An agent is called rational, if agent *i* always tries to maximize its utility u_i by finding its best strategy.

The system wide goal in mechanism design is defined by a social choice function *g*(), which, given agent types, selects the optimal outcome. Give mechanism with outcome function *o*(), we say that a mechanism implements social choice function *g*() if the outcome computed with equilibrium agent strategies is a solution to the social choice function for all possible agent preference. An output function *o*() of a mechanism is allocatively-efficient if it maximizes the summation of valuations of all agents, i.e., $\sum_{i=1}^{n} v_i(t_i, o) \geq \sum_{i=1}^{n} v_i(t_i, o')$ for all possible types *t*. A mechanism is efficient if it implements an allocatively-efficient social choice function.

A direct-revelation mechanism is a mechanism in which the only actions available to agents are to make direct claims about their preferences v_i to the mechanism. An incentive compatible mechanism is a direct-revelation mechanism in which agents report their valuations v_i to the mechanism truthfully so as to maximize its utility. Incentive-compatibility captures the essence of designing a mechanism to overcome the self-interest of agents: in an incentive compatible mechanism an agent will choose to report its private information truthfully in order to maximize its utility. A direct-revelation mechanism is strategyproof if truth-revelation is dominant strategy equilibrium.

There is a class of strategyproof mechanisms, called the VCG by Vickrey [8], Clarke [9], and Groves [10] . It utilizes the second price sealed bid (SPSD) auctions to compute the payment: $p_s\{a_i = \infty\} - p_s\{a_i = 0\}$. With the SPSD auctions, the mechanism is truth-telling so each agent will reveal its real type based on its own profit. Thus, the aggregation of decisions of every selfish agent will result in a global optimal solution.

3 Truthful Mechanism Design

In this section, we analysis the two scenarios introduced in the second section in detail, and design truthful mechanisms to resolve the selfish problems in the Sensor Web.

3.1 One Sensor and Several Users

In this scenario, only one sensor is available and several users want to utilize this sensor. User has to pay for utilizing the sensor and the sensor would like to maximize its benefit. Like in Figure 1, one camera supplies service to *n* users. And we further assume that only one user can use the sensor at a time. E.g., a user wants to observe some places or something using the sensor, while he does not want others know what he is watching because of some private reasons. So the using of sensor is exclusive.

Cost of Sensor

In the traditional mechanism design, it usually only considers the price that each user would like to bid. E.g., in the Second Price Sealed Auction, the user who bid the highest price will win the sensor's service and will pay the second highest price bid by other user. In this case, the cost of the sensor is neglected or considered to be equal for each user. However, sometimes the cost cannot be neglected because it is comparable with the price bided or is not equal for each user. Take the camera for example, when supplying service to users, the camera might be required to turn around in order to get the image in a certain derection. The angle need to turn is usually different for each user. If the cost of turning around is related with the angle and comparable with the price bided by user, then the cost of sensor has to be considered in the mechanism.

In this paper, we assume that the sensor's cost C_i for serving user *i* is caused by turning around and linear with the turning angle α_i :

$$C_i = C_0 + k\alpha_i \quad (0 \le \alpha_i \le 180) \tag{1}$$

Here C_0 is the fixed cost of sensor, k is a constant parameter and α_i is the turning angle. In the scenario of one sensor and several users, we assume that the formula above is a common knowledge for every user.

Type of User

As is shown in Figure 1, each user *i* has its type V_i , which is the value it can obtain for using the sensor, and it also equals to the price that user *i* would like to pay for using the sensor.

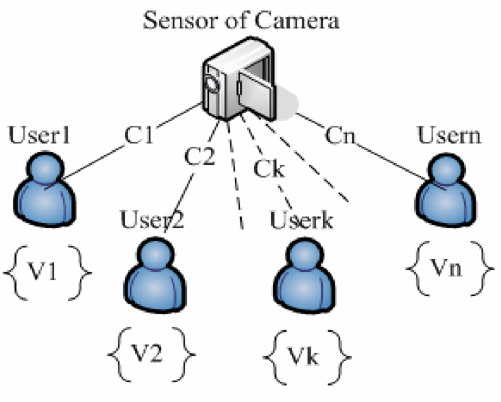


Fig. 1. Scenario of one sensor and several users

The sensor's value may not be the same for different users, and V_i is the private information which is only known by user *i*. In this paper, we assume that V_i is linear with the angle α_i that the sensor turns around for user *i*:

$$\begin{cases} V_i = V_{0i} + g_i\alpha_i \quad (0 \le \alpha_i \le \alpha_{0i}) \\ V_i = V_{0i} + g_i\alpha_{0i} - g_i(\alpha_i - \alpha_{0i}) \quad (\alpha_{0i} \le \alpha_i \le 180) \end{cases} \tag{2}$$

Here V_{0i} is the fixed benefit the user can get from the sensor, g_i is a constant parameter and α_{0i} is the angle which can maximize the user's type.

One purpose of our mechanism is to make sure that every user tell their V_i truthfully, i.e., tell the truthful V_{0i}, g_i, α_i and α_{0i}.

With the knowledge of cost of sensor and type of user, we next discuss the one sensor and several users situation in detail.

In this situation, only one user can control the sensor at a time. Every user would like to obtain the sensor in a low price, and sensor wants to get as much pay back as possible to maximize its revenue. So we need to design a mechanism to make sure that every user tells its true type and at the same time sensor could also maximize its revenue. Our mechanism is divided into two phase: sensor assignment and payment calculation.

Sensor assignment

User *i* gives its type V_i calculated by V_{0i}, g_i, α_i and α_{0i}, and needs sensor turning α_i in order to point to the proper direction, which will cause the turning cost $C_i = C_0 + k\alpha_i$. Then the value V_i - C_i which stands for the pure benefit of sensor can be calculated, and the user with the highest V_i - C_i wins the sensor.

Payment calculation

Assume V_j - C_j of user *j* is the second highest value compared with the V_i - C_i of user *i*, then the winner user *i* should pay $V_j - C_j + C_i$ to the sensor.

Lemma 1. The payment of user *i* $V_j - C_j + C_i$ is always lower than his type: V_i

Proof. $V_i - C_i > V_j - C_j \Rightarrow V_j - C_j + C_i < V_i$

This lemma demonstrates that the payment of user *i* is always lower than his true type, so every user will participate the mechanism voluntarily.

Theorem 1. The mechanism described above is strategyproof

Proof. First of all, $C_i = C_0 + k\alpha_i$, the formula is common knowledge for every user, so C_i and C_j will be calculated truthfully.

If a lower value of V_i is claimed by user *i*, $V_i - C_i$ will be lower than its reality. If the truthful value of $V_i - C_i$ is the highest among all users, claiming a lower V_i might make it not be the highest value any more and then cannot obtain the sensor. If the truthful value of $V_i - C_i$ is not the highest, then claming a lower V_i will make the value of $V_i - C_i$ lower and still cannot obtain the sensor. So the user has no incentive to claim a lower V_i.

Alternatively, if a higher value of V_i is claimed by user *i*, $V_i - C_i$ will be higher than its reality. If the truthful value of $V_i - C_i$ is the highest among all users, claiming a higher V_i will not change its payment. If the truthful value of $V_i - C_i$ is not the highest, e.g., $V_i - C_i < V_j - C_j$, then claiming a higher V_i might make the user obtain the user, but his payment $V_j - C_j + C_i$ will be higher than his type V_i, which cannot be accepted by the user. So the user has no incentive to claim a higher V_i. Thus truth-telling is a dominant strategy for this mechanism, i.e., our mechanism is strategyproof.

3.2 One User and Several Sensors

In this scenario, several sensors would like to supply service to the user and only one user need the service. And we further assume that the user only needs one sensor at a time. E.g., when several cameras can supply the video of the same place, user only needs to select one sensor according to the price, quality of the video etc.

Type of Sensor

As is shown in Figure 2, each sensor *i* has its Type C_i, which is the cost when it supply service to user and it also equals to the price that the sensor want the user to pay. The cost of different sensor might be different, and C_i is private information for every sensor. C_i can be calculated using the similar one with formula (1) as is shown in the one sensor and several users scenario:

$$C_i = C_{0i} + k\alpha_i \quad (0 \le \alpha_i \le 180) \tag{3}$$

Here C_{0i} is the fixed cost of sensor *i*, other parameters' meaning is same as before.

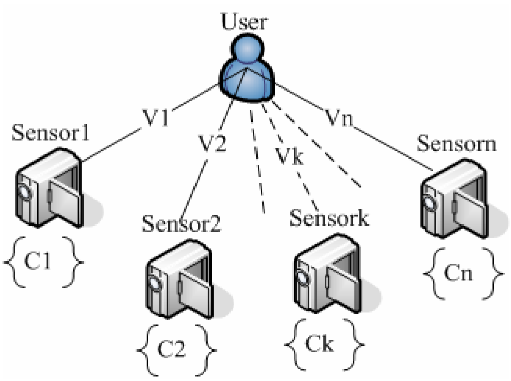


Fig. 2. Scenario of one user and several sensors

Sensor's Value in User's Perspective

In this scenario, different sensor may have different value for the user, so we can also utilize the same formula (2) in the former part to calculate V_i. Here we assume that this formula is common knowledge for every sensor.

With the knowledge of cost of sensor and type of user, we next discuss the one user and several sensors situation in detail.

In this situation, user only needs one sensor. Every sensor would like to supply service in a high price, while user wants to get the service in a low price. We design a mechanism as below to make sure that every sensor gives its true type and maximize user's benefit.

User Assignment

Sensor *i* gives its type C_i, and calculates its value V_i to user according to formula (2). Then sensor with lowest $C_i - V_i$ win and will supply service to user.

Payment Calculation

Assume $C_j - V_j$ is the second lowest value compared with $C_i - V_i$, Then the user will pay $C_j - V_j + V_i$ to sensor *i*.

Lemma 2. The payment to sensor *i* $C_j - V_j + V_i$ is always higher than his type: C_i

Proof. $C_i - V_i < C_j - V_j \Rightarrow C_j - V_j + V_i > C_i$

Theorem 3. The mechanism described above is strategyproof

Proof. First of all, $V = k\alpha$, k is a constant parameter, the angle away from the baseline will be given truthfully, because the user can easily find the cheating when the angle it gets is different from the sensor's claim. So V_i and V_j will be truthfully calculated.

If a higher value of C_i is claimed, $C_i - V_i$ will be higher than its reality. If the truthful value of $C_i - V_i$ is the lowest among all sensors, claiming a higher C_i might make it not be the lowest value any more and then cannot win the user. If the truthful value of $C_i - V_i$ is not the lowest, then claming a higher C_i will make the value of $C_i - V_i$ higher and still cannot win the user. So the sensor has no incentive to claim a higher C_i .

Alternatively, if a lower value of C_i is claimed, $C_i - V_i$ will be lower than its reality. If the truthful value of $C_i - V_i$ is the lowest among all sensors, claiming a lower C_i will not change its revenue. If the truthful value of $C_i - V_i$ is not the lowest, e.g., C_i - V_i > C_j - V_j , then claiming a lower C_i might make the sensor win the user, but his revenue C_j - $V_j + V_i$ will be lower than his type C_i , which cannot be accepted by the sensor. So the user has no incentive to claim a lower C_i . Thus truth-telling is a dominant strategy for this mechanism. I.e., our mechanism is strategyproof.

4 Simulation Result

In this section, we will give some simulation results of our mechanism. First of all, we demonstrate the result of scenario with one sensor and several users. Table 1(a) gives the data we use in the simulation of one sensor and several users. Applying our mechanism on these data, we could calculate the payment of user and the pure profit of sensor. The pure profit of sensor equals to its total profits got from the users minus it cost of supplying service to users.

As is shown in Figure 3(a), user four will get the sensor when not considering the cost of sensor, and user three will get the sensor when considering the cost of sensor. What's more, the payment of user three is less than user one, which is good news in the user's perspective.

Table 1. Scenario data (a) one sensor and several users (b) one user and several sensors

(a)

No. i	1	2	3	4
α_{0i}	10	20	30	40
C_0	1			
k	0.3			
C_i	4	7	10	13
V_{0i}	10	15	30	22
g_i	0.4	0.5	0.5	0.6
V_i	14	25	45	46
$V_i - C_i$	10	18	35	33

(b)

No. i	1	2	3	4
α_{0i}	30			
C_0	1	2	3	6
k	0.1	0.2	0.3	0.3
C_i	4	8	12	15
V_{0i}	10	15	30	22
g_i	0.4	0.5	0.5	0.6
V_i	14	25	45	46
$C_i - V_i$	-10	-17	-32	-31

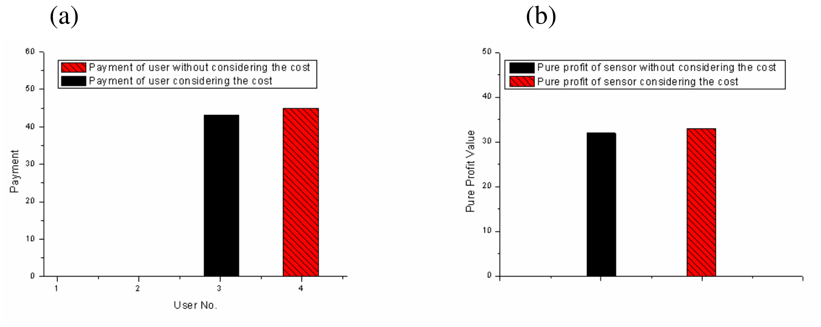


Fig. 3. (a) Payment of users with one sensor and several users (b) Pure profit of sensor with one sensor and several users

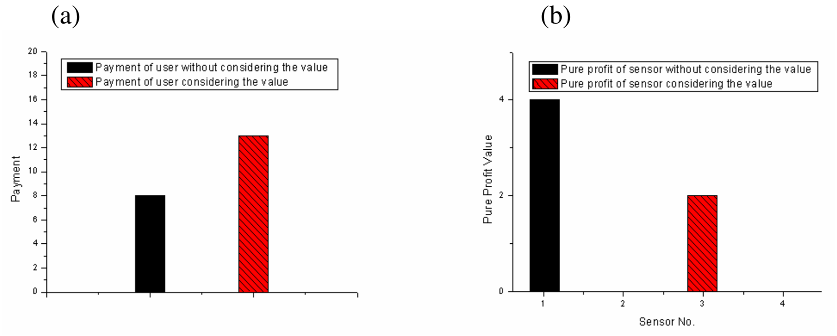


Fig. 4. (a)Payment of users with one user and several sensors (b) Pure profit of sensor with one user and several sensors

Figure 3(b) demonstrates the pure profit of sensor. From the graph, we can see that although the payment of user when considering the cost is lower than the one when not considering the cost, the pure profit of sensor is relative high. It is good news for both users and the sensor, because the user can pay less while the sensor can get more.

Table 1(b) gives the data we use in the simulation of one user and several sensors. Applying our mechanism on these data, we could calculate the payment of user and the profit of sensor.

As is shown in Figure 4(b), sensor one will win the user when not considering the value of user, and sensor three will win the user when considering the value of user.

Figure 4(a) demonstrates the payment of user. From the graph, we can see that payment give by user who considers the value is higher than the one not considering.

5 Conclusion

The selfish problems of Sensor Web are described and classified into two scenarios in this paper. And each of the scenarios is further classified into two situations. The variable cost and value issues which are quite practical are also introduced to the sensor web. Four corresponding truthful mechanisms are also designed to make sure that user or sensor can tell its true type and get maximum benefits, so the Sensor Web is guaranteed running smoothly under these mechanisms. The simulation results proved that our mechanism is truthful and quite efficient for both sensor and user.

Next we would like to expend the mechanism to the scenario of several sensor and several users, which is more complex and challenging.

Acknowledgment

This research was partially supported by Natural Science foundation of China under grant No. 60573138 and National Basic Research Program of China (973 program) under grant No. 2006CB303000.

References

[1] Delin, K.A.: The Sensor Web: A Macro-Instrument for Coordinated Sensing. Sensors, pp. 270–285 (July 2002)

[2] http://research.microsoft.com/nec/senseweb/

[3] http://atom.research.microsoft.com/sensormap/

[4] Rogers, A., Dash, R.K., Jennings, N.R., Reece, S., Roberts, S.: Computational Mechanism Design for Information Fusion within Sensor Networks. In: Proceedings of Ninth International Conference on Information Fusion (2006)

[5] Wu, M.Y., Shu, W.: InterSensorNet: strategic routing and aggregation. In: Global Telecommunications Conference (2005)

[6] Nisan, N., Ronen, A.: Algorithmic mechanism design. Games and Economic Behavior 35, 166–196 (2001)

[7] Papadimitriou, C.H.: Algorithms, games, and the internet. In: Proceedings of the 33rd Symposium on Theory of Computing (2001)

[8] Vickrey, W.: Counterspeculation, auctions and competitive sealed tenders. Journal of Finance, 8–37 (March 1961)

[9] Clarke, E.H.: Multipart pricing of public goods. Public Choice, 17–33 (1971)

[10] Groves, T.: Incentives in teams. Econometrica, 617–631 (1973)

An Enhanced DV-hop Localization Algorithm for Irregularly Shaped Sensor Networks [⋆]

Yanchao Niu, Sidong Zhang, Xiaoyu Xu, Hongwei Huo, and Shuai Gao

Beijing Jiaotong University, 100044 Beijing, China
gniux819@gmail.com, {sdzhang, 06120186, hwhuo, shgao}@bjtu.edu.cn

Abstract. Position information in sensor networks is crucial for many applications. A large number of localization algorithms have been developed, however, most of them need special measurement tools and could not work well in irregularly shaped sensor networks. In this paper, we present an Enhanced DV-hop Localization Algorithm (EDLA) to deal with irregularly shaped network topology. With the same process of DV-hop, a reference node of EDLA can delimit approximate regular shaped regions as convex hulls. Based on the inclusion test within convex hulls, the expected average per-hop distance information is disseminated by qualified relay nodes and used for position estimation. Simulation results show that the proposed localization method outperforms DV-hop and PDM(proximity-distance map) in irregularly-shaped sensor networks if both the localization error and computational complexity are considered.

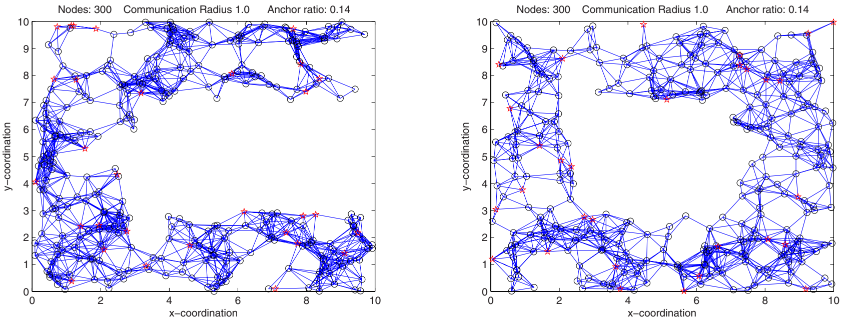
1 Introduction

Wireless sensor networks(**WSN**) have shown great promise in recent years with the development of MEMS(Micro Electromechanical System) technology. They are typically composed of large-scale, resource-constrained sensor nodes which have the ability of communication with each other and cooperatively collect simple sensing information from the physical world. It is crucial for sensor data combined with spatial information in many domains such as environmental monitoring, smart building failure detection and military target tracking. The position information of sensor also helps to facilitate routing to the geographic fields of interests, and to provide location-based services like determining the quality of coverage and achieving load balance.

Localization has become an essential aspect in **WSN**. GPS (Global Positioning System) is the most well-known public location service. However, the expensive and huge energy consumption properties make it impractical if every sensor node is equipped with GPS receiver. Instead, it is assumed that only a small portion of sensor nodes are aware of their position information by GPS or manual configuration in most localization methods. We call these nodes *anchor nodes* and other nodes, called *unknown nodes*, can estimate their positions by *anchor nodes*.

[⋆] This research is supported by the National Science Foundation of China (No.60572037).

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 694–704, 2007.



(a) C-shape (b) O-shape

Fig. 1. Irregularly shaped topology

The topology of a network plays an important role in the positioning process. Most research work including [1] [2] [3] [4] [5] assumes that sensor nodes have been uniformly deployed and the network topology is regularly-shaped. Unfortunately, the assumption does not hold in practice due to irregularly shaped of the region, different terrain conditions, etc. C-shaped and O-shaped topology are two of the most representative irregular topologies in Figure 1. As a result, most of algorithms proposed above cannot work well in this irregularly-shaped network. For example, the average position error of DV-hop, which is one of the pioneering method without any range measurement tools, reaches 1.7696 (in unit of communication radius) in C-shaped topology and 1.7810 in O-shaped topology compared with 0.9739 in regularly-shaped topology.

In this paper, we present an enhanced DV-hop localization algorithm (EDLA) to deal with irregularly shaped network topology. With simple connectivity information, a reference node of EDLA can delimit approximate regular shaped regions as convex hulls. Based on the inclusion test within convex hulls, the expected average per-hop distance information is disseminated by qualified relay nodes.

The rest of this paper is organized as follows. In section 2, we present related work in this area of sensor network localization. In section 3, we describe the detail of EDLA algorithm. Following that, section 4 presents qualitative comparison of several experiments and we conclude the paper in section 5 .

2 Related Work

Ad hoc positioning system (APS) [1] is one of the earliest distributed localization algorithms and guides *unknown nodes* to find the distances to *anchor nodes*. Three methods are proposed: DV-hop, Dv-distance and Euclidean. Among them, DV-hop has attract more attention because connectivity and hop-count values rather than any range measurement tools are involved. In DV-hop, each *anchor nodes* flood their global positions to entire network and every nodes keep their positions and minimum hop-count values. After *unknown nodes* receive the average hop distance which is calculated and propagated by *anchor nodes*, they convert hop counts to distances and perform triangulation to estimate their own positions. The formula of average hop distance c_i is Equation 1.

$$c_i = \frac{\sum \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum h_{ij}} \qquad i \neq j \tag{1}$$

In Equation 1, $(x_i, y_i)_{i=0,...}$ is the position of *anchor node i* and h_{ij} is the minimum hop-count values. DV-hop works well in regularly shaped network where nodes are randomly deployed with a uniform distribution and average hop distance can be considered as a constant. However, in irregularly shaped network, average hop distance varies greatly which causes enormous error. Savarese et al. have presented an algorithm [2] with iterative refinement to improve the localization accuracy and R. Nagpal et al. have proposed an *Amorphous* scheme [3]which use improved formula instead of anchor-to-anchor measurements to calculate the average hop distance. However, neither of them could do well in irregularly shaped network.

APS does not consider the issue of *anchor nodes* selection and Savvides et al. propose a Maximum Likelihood Estimate of node positions using adequate available *anchor nodes* [6]. Cheng et al. [7]show that using all *anchor nodes* may not yield accuracy position in irregularly shaped network and develop a *anchor nodes* selection strategy called *APS(Near-3)*. However, selecting the nearest 3 anchors may not always obtain the best position estimation and need other technique to deal with the variance of average hop distance.

Shang et al. propose a MDS-based centralized localization method [8], named *MDS-MAP*. However, it needs global connectivity information for all sensor nodes and does not work well in irregular or large-scale networks. Shang et al. later develop a new algorithm [9] called *MDS-MAP(P)*, which establishes local maps using MDS and merge them to construct a global map. *MDS-MAP(P)* considers the local map within 2-hop or 3-hop distance as regular topology and it requires much more computation cost and suffers from the error propagation in merging phase.

Lim and Hou propose linear mapping method [10], called *proximity-distance map* (PDM),which transform proximity measurement like hop counts or estimated path distances between sensor nodes into geographic distances. The technique, truncated singular value decomposition (SVD), can reduce the effect of irregularly shaped topology and obtains accurate position estimates. Although PDM uses hop-counts as proximities and its operation is similar to DV-hop without any range measurement, truncated SVD also requires huge computation complexity and memory cost which is hard for sensor nodes.

3 EDLA Design Details

Considering the tradeoff between accuracy and computational complex, in this section, we propose an enhanced DV-hop localization algorithm (EDLA) in detail. EDLA uses only connectivity information which is similar to DV-hop, and every *anchor node* delimits approximate regularly shaped regions as convex hulls. *unknown nodes* estimate their position with proper *anchor nodes* based on inclusion test inside convex hulls.

The placement strategy of *anchor nodes* is an importance factor of localization performance. We assume that *anchor nodes* are randomly deployed in the network with a uniform distribution and communication range of each *anchor node*, denoted as r, which is a circle centered at the *anchor node*. We also consider a connected network where each node has at least one neighbor.

There are three main issues in our distributed algorithm that need to be examined:

- How can an *anchor node* detect approximate regularly shaped region? —Regular region detection
- How can an *anchor node* delimit the regular region considering accuracy and computational complex? —Convex hulls construction
- How can an *unknown node* obtain proper anchors information within regular region? —Inclusion test for *unknown node*

3.1 Regular Region Detection

Similar with conventional DV-hop localization algorithm, every *anchor node* floods its position information to the whole network and all sensor nodes can obtain minimal hop-count values to every *anchor node*. In this way, *anchor node* i with position (x_i, y_i) gets hop-count measured hop_{ij} to *anchor node* j with position (x_j, y_j) and geographic distance $dist_{ij}$ calculated by $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$. Then $hopdist_{ij}$, average hop distance to *anchor node* j, is the ratio of $dist_{ij}$ and hop_{ij}. Depending on $hopdist_{ij}$, *anchor node* i can determinate whether there is a straight line between i and j or not.

Criterion Rule 1. *Let k be a coefficient about tolerated hops. For any anchor node i and j, if $hopdist_{ij}$ is more than $r(1 - \frac{k}{hop_{ij}})$, we say anchor node i detects that j lies in the regularly shaped region of i.*

Proof. Let there be an obstacle between i and j in Figure 2 and the hop-count value in straight line be less than Δh ($\Delta h > k$) of hop_{ij} with the obstacle, i.e. $hop_{ij} - \Delta h$.

In Figure 3, the straight-line distance between i and j $dist_{ij}$ can be expressed as $r * n - \Delta d$. Therefore,

$$\begin{aligned} hopdist_{ij} &= \frac{dist_{ij}}{hop_{ij}} = \frac{r * n - \Delta d}{hop_{ij}} \\ &= \frac{r * (hop_{ij} - \Delta h) - \Delta d}{hop_{ij}} \\ &< \frac{r * (hop_{ij} - \Delta h)}{hop_{ij}} \\ &= r * (1 - \frac{\Delta h}{hop_{ij}}) \end{aligned}$$

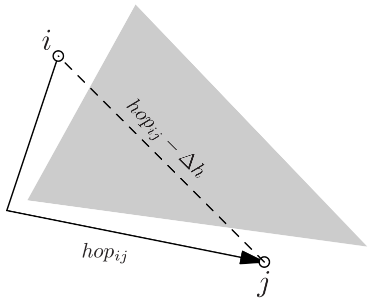


Fig. 2. Criterion Rule diagram 1

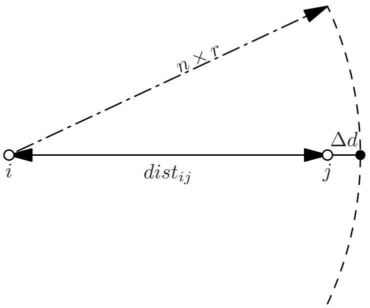


Fig. 3. Criterion Rule diagram 2

Because $\varDelta h$ exceeds tolerated hops k, $hopdist_{ij}$ will be less than $r * (1 - \frac{k}{hop_{ij}})$.

Depending on the dynamic threshold $r(1 - \frac{k}{hop_{ij}})$, *anchor node* i regards the set, called *regular anchors set* where *anchor nodes* satisfy Criterion Rule 1 as its regularly-shaped region.

3.2 Convex Hulls Construction

After the anchor i forms the *regular anchors set*, it needs to delimit the set border as the regularly-shaped region. We use the "divide and conquer" method and the reasons are as follows:

– If the border shape of the whole *regular anchors set* is reentrant polygon, some *anchor nodes* is not appropriate for all *unknown nodes* within the reentrant polygon. A simulation example in Figure 4 where the right part and down part of *anchor node* 1 should be split and each of them is a regular region.
– If we use all triangular combinations as regularly shaped regions, high computation complexity will be required. There will be $\frac{N(N-1)}{2}$ possible regions which N is the size of the *regular anchors set.*

For efficiency and validity, we divide the *regular anchors set* by angle A_θ. The value A_θ affects the amount of computation as well as the quality. We set $A_\theta = \frac{\varPi}{4}$ in the experiments. The steps of "divide and conquer" method by *anchor node* i are as follows:

1. Divide several region sets $\{R_{\langle A_\theta,k\rangle}(\theta_{\rho_{max}})\}_{k=1,2,...}$. For the whole *regular anchors set*, do the following:
   (a) Construct polar coordinates (ρ, θ) for *regular anchors set* with the center of *anchor node* i. According to the cartesian coordinates of *anchor node* i and others in *regular anchors set*, the polar coordinates set is calculated and denoted as $P = \{(\rho_n, \theta_n)\}_{n=1,2,...,N}$

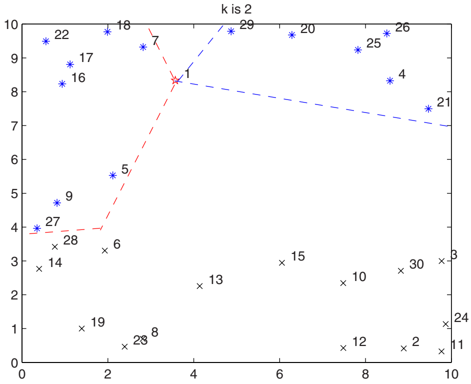


Fig. 4. The anchor node 1 detect the *regular anchors set* at its right side and down side in C-shaped network. The red $\star$, blue $*$ and gray $\times$ in figure are the position of node 1, *anchor nodes* matched the Criterion Rule 1 and *anchor nodes* unmatched the Criterion Rule 1 respectively.

(b) Select the *anchor node* with the maximum polar angle.

$$\widehat{n} = \arg\max_{n} \theta_n \tag{2}$$

We set $\theta_{\rho_{max}} = \theta_{\widehat{n}}$, and remove $(\rho_{\widehat{n}}, \theta_{\widehat{n}})$ from polar coordinates set P.

(c) Select the nodes whose angle among $\theta_{\rho_{max}}$ is less than A_θ

$$R_{\langle A_\theta, k\rangle}(\theta_{\rho_{max}}) = \{(\rho_v, \theta_v) \in P| \mid \theta_{\rho_{max}} - \theta_v \mid < A_\theta\}$$

We also remove these nodes' polar coordinates from P.

(d) Do the same process above until the polar coordinates set P is empty.

2. For each region set $\{R_{\langle A_\theta, k\rangle}(\theta_{\rho_{max}})\}_{k=1,2,...}$, we check whether two θ have the same value which means the two nodes are collinear with *anchor node* i. If any pair exists, the node with the smaller ρ will be deleted from the region set.
3. The region set with isolated node should be merged into its neighbor set in which the angle between the isolated node's θ and other $\theta_{\rho_{max}}$ is smallest. Then we denote the adjustment region sets as $\{\widehat{R_{k'}}\}_{k'=1,2,...}$
4. Construct convex hull by every $\widehat{R_{k'}}$. Convex hull of $\widehat{R_{k'}}$ is the enclosing convex polygon with smallest area and any *unknown nodes* within the convex hull can use these *anchors nodes* for position estimation which is the same as DV-hop process in regular region. In this paper, we use a simple Graham's algorithm to construct convex hull [11]. Firstly, we sort *anchors nodes* of every $\widehat{R_{k'}}$ by θ in ascending order, which are labeled $L_{k'} = \{l_{ik'}\}_{i=1,2,...,\|R_{k'}\|}$. Then selecting the *anchors nodes* i as starting point and *anchors nodes* in $L_{k'}$ by order, the continuous four nodes are determined whether the sequence formed by the four nodes is a convex chain. If true, the process continues;

otherwise, the third node will be deleted from $L_{k'}$ and the determination goes on. Finally, the *anchors nodes* leave in $L_{k'}$ and *anchors nodes* i are the extreme points of a convex hull. The criterion rule for the convex chain is simple in which the first node and fourth node locate in the same side of the line determined by the second node and the third node. We summarize the process with pseudocode in Algorithm 1.

Algorithm 1. Convex Hull Algorithm: Graham

```
Sort all points in $\widehat{R_{k'}}$ by $\theta$ in ascending order, denoted as $L = \{l_1, l_2, \cdots, l_n\}$
$l_0 = $ anchor node $i$
Stack $S = (l_0, l_1, l_2) = (l_{top-2}, l_{top-1}, l_{top})$; top indexes stack top
$i \Leftarrow 3$
while $i < n$ do
    if $l_{top-2}$ and $l_i$ locate in the same side of $\overline{l_{top-1}l_{top}}$ then
        PUSH($S$, $l_i$)
        $i \Leftarrow i + 1$
    else
        POP($S$)
    end if
end while
Output $S$ as the extreme points of the convex hull
```

3.3 Inclusion Test for *Unknown Nodes*

Unknown nodes need to detect whether they are inside a convex hull so that they can determine which *anchor nodes* are qualified for localization. The inclusion test for an *unknown node* is simple since it can get every *anchor node*'s position information and corresponding hop-count value in convex hull vertices by the first phase of *anchor nodes*' dissemination. According to the procedure of convex hull construction above, a convex hull consists of finite non-collinear *anchor nodes* and can be divided into several triangles with the common vertex by the start *anchor node*. Once the *unknown node* detects inside one of these triangles, it will pass the inclusion test. The triangular inclusion test is Criterion Rule 2.

Criterion Rule 2. *When a point lies inside a triangle, the sum of areas enclosed by the point and three triangular vertices is equal to the triangular area, otherwise it will be larger than the triangular area.*

An example shows in Figure 5. The *unknown node* x calculates the distance to $a_1, a_2, \cdots, a_5$ by the hop-count values and the average hop distance of start point a_1. Subsequently, x checks the Criterion Rule 2 for each triangle. For example, in triangle $\triangle a_1a_3a_4$ of Figure 6, x and a_1, a_3, a_4 enclose three triangle $\triangle a_1xa_4$, $\triangle a_1xa_3$,$\triangle a_3xa_4$. The triangular area is calculated by $S = \sqrt{p(p-l_a)(p-l_b)(p-l_c)}$, where l_a, l_b, l_c are triangle sides and p is

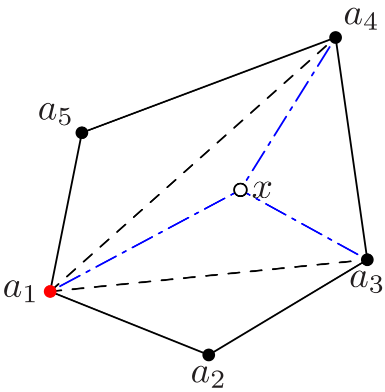


Fig. 5. An inclusion test example for an *unknown node* x. $a_1, a_2, \cdots, a_5$ are *anchor nodes* and delimit a convex hull constructed by a_1.

$\frac{1}{2}(l_a + l_b + l_c)$. It is obvious that the sum areas of $\triangle a_1 x a_4$, $\triangle a_1 x a_3$ and $\triangle a_3 x a_4$ are equal to the areas of $\triangle a_1 a_3 a_4$ when x locates inside the $\triangle a_1 a_3 a_4$, and when x' locates outside the $\triangle a_1 a_3 a_4$, $S_{a_1 x' a_4} + S_{a_3 x' a_4} + S_{a_1 x' a_3}$ are larger than $S_{a_1 a_3 a_4}$

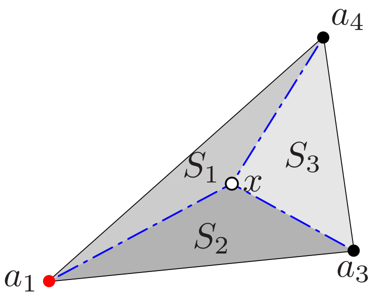


Fig. 6. x lies inside the triangle

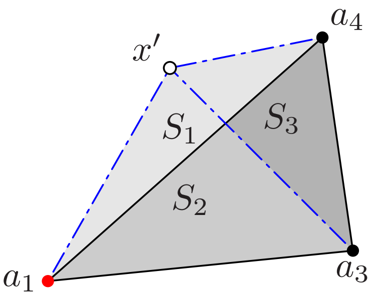


Fig. 7. x' lies outside the triangle

4 Simulation Result

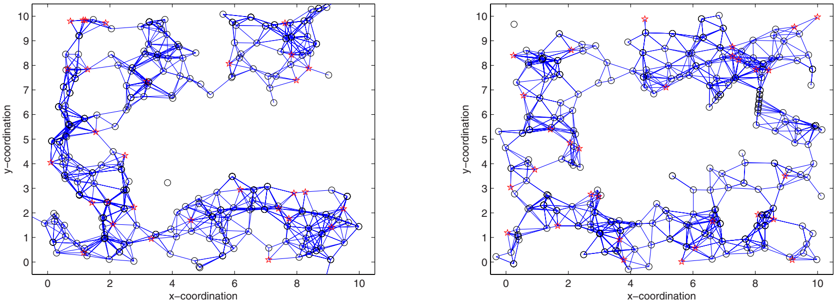
To evaluate the performance of EDLA, we have conducted a simulation in C++ environment and compare the performance of DV-hop, PDM with our EDLA method in different irregularly shaped network. The network parameters list in Table 1.

In each experiment of C-shaped network and O-shaped network, the position error is calculated by Equation 3, in which N is the number of *unknown nodes*, (x_i, y_i) and (x'_i, y'_i) are the true and estimated position coordinates of *unknown nodes* i.

$$\xi_p = \frac{\sum_{i=1}^{N} \sqrt{(x'_i - x_i)^2 + (y'_i - y_i)^2}}{N \times r} \tag{3}$$

Table 1. Simulation setup

Parameters	Scope of values
Number of nodes	300
Rectangular field of sensor network	$10r \times 10r$
Network topology	O-shaped, C-shaped
Communication radius	$1.0r$
Ratio of *anchor nodes*	from 4% to 14%



(a) The C-shape result of EDLA

(b) The O-shape result of EDLA

Fig. 8. The resulting topology of EDLA in C-shaped network and O-shaped network

Compared with the original topologies in Figure 1(a) and Figure 1(b), Figure 8(a) and Figure 8(b) show the resulting topology of EDLA, which have 300 sensor nodes with 30 *anchor nodes* and each node has the same communication radius 1.0. The '$*$' denotes the position of an *anchor node* and '$\circ$' is an *unknown node*. From the two figures, we can see that EDLA is capable of doing well in irregularly-shaped sensor networks.

Subsequently, we investigate the effect of the ratio of *anchor nodes* on the localization accuracy in C-shaped sensor networks and O-shaped sensor networks. Although EDLA does not perform better than PDM when the ratio of *anchor nodes* is more than 0.07 in Figure 9, the DV-hop works worst in C-shaped networks. In O-shaped networks of Figure 10, EDLA gives even the best localization accuracy in all the three algorithms.

In terms of computational complexity, PDM uses the singular-value decomposition (SVD) technique to obtain the proximity-distance map which needs huge memory cost and has computational complexity of $O(n^3)$ as the parameters n of the number of *anchor nodes*. However, the EDLA only needs to check *anchor nodes* which meet the condition of Criterion Rule 1 and construct convex hulls. Its computational complexity is only $O(n)$.

In summary, EDLA works better in irregularly-shaped sensor network than Dv-hop and it is effective to use qualified *anchor nodes* and average hop distance

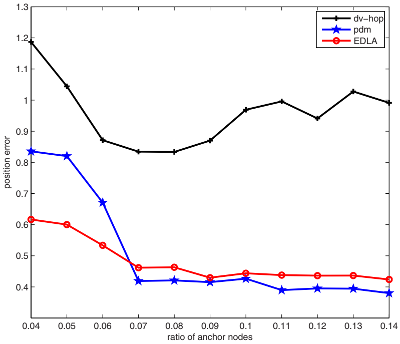


Fig. 9. The localization accuracy in C-shaped network

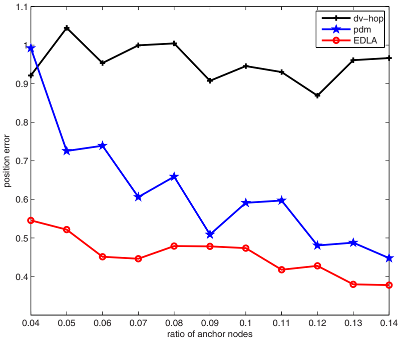


Fig. 10. The localization accuracy in O-shaped network

for localization. The performance of EDLA is comparable with PDM but has less computational complexity.

5 Conclusion

In this paper, we have designed an enhanced Dv-hop localization algorithm (EDLA) to deal with the irregularly-shaped sensor networks. With the same process of Dv-hop, we derive the criterion rule and detect anchor set in regular region for one *anchor node*. We construct convex hulls as the border of approximate regularly shaped region. Any *unknown nodes* inside the convex hulls can use the *anchor node* information to estimate its position. Finally, the simulation results prove that the EDLA reduces the computational complexity while providing decent localization accuracy for irregularly-shaped networks. The deployment of *anchor nodes* is an important factor in localization performance and should brought into consideration especially when the deployment changes, which demands further investigation.

References

1. Niculescu, D., Nath, B.: Dv based positioning in ad hoc networks. Telecommunication Systems 22(14), 267–280 (2003)
2. Savarese, C., Rabay, J., Langendoen, K.: Robust positioning algorithms for distributed ad-hoc wireless sensor networks. In: Proc. USENIX Technical Ann. Conf. (2002)
3. Nagpal, R., Shrobe, H., Bachrach, J.: Organizing a global coordinate system from local information on an ad hoc sensor network. In: Zhao, F., Guibas, L.J. (eds.) IPSN 2003. LNCS, vol. 2634, pp. 333–348. Springer, Heidelberg (2003)
4. He, T., Huang, C., Blum, B., Stankovic, J., Abdelzaher, T.: Range-free localization schemes for large scale sensor networks. In: MobiCom 2003. Proceedings of the 9th annual international conference on Mobile computing and networking, pp. 81–95 (2003)

5. Wang, S.-S., Shih, K.-P., Chang, C.-Y.: Distributed direction-based localization in wireless sensor networks. Computer Communications 30(6), 1424–1439 (2007)
6. Savvides, A., Han, C.-C., Srivastava, M.: Dynamic fine-grained localization in adhoc networks of sensors. In: Proc. 7th Ann. Intl. Conf. on Mobile Computing and Networking, pp. 166–179 (2001)
7. Cheng, K.-Y., Tam, V., Lui, K.-S.: Improving aps with anchor selection in anisotropic sensor networks. In: Autonomic and Autonomous Systems and International Conference on Networking and Services, pp. 49–49 (2005)
8. Shang, Y., Ruml, W., Zhang, Y., Fromherz, M.P.J.: Localization from mere connectivity. In: MobiHoc 2003: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing, pp. 201–212 (2003)
9. Shang, Y., Ruml, W.: Improved mds-based localization. In: INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies, vol. 4, pp. 2640–2651 (2004)
10. Lim, H., Hou, J.C.: Localization for anisotropic sensor networks. In: INFOCOM 2005. Twenty-fourth Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 1, pp. 138–149(2005)
11. O'Rourke, J.: Computational Geometry in C, 2nd edn. Cambridge University Press, Cambridge (1988)

Model for Survivability of Wireless Sensor Network

Xianghui Liu[1], Jing Ning[1], Jun Li[1], Jianping Yin[2], and Ming Li[2]

[1] School of Electronic Science & Engineering, National University of Defense Technology,
Changsha City, Hunan Province, 410073, China
[2] School of Computer Science, National University of Defense Technology,
Changsha City, Hunan Province, 410073, China
LiuXH@tom.com

Abstract. In this paper we focus on the survivability of wireless sensor networks, with particular emphasis on industries requirements. we develop a model to evaluate the tradeoffs between the cost of defense mechanisms for Wireless Sensor Network and the resulting expected survivability after a network attack. The model consists of three parts. The first part simulates the occurrence of attacks or incidents. The second part simulates the impact of an attack on the WSNS which depends on the type of attack and the defense mechanism installed in the network. And the third part assesses the survivability of the system which depends on the degree of its degradation after the attack. We demonstrate through simulation the model's effectiveness in mitigating attacks.

1 Introduction

Wireless Sensor networks (WSNSs) are integration of sensor techniques, nested computation techniques, distributed computation techniques and wireless communication techniques. They can be used for testing, sensing, collecting and processing information of monitored objects and transferring the processed information to users. Sensor network is a new research area of computer science and technology and has a wide application future. Both academia and industries are very interested in it [1].

Traditionally, networks have strongly relied on physical security. The concept of a network firewall is an example of this approach. A firewall is intended to provide an access control division between the insecure public network (the Internet) and the seemingly secure private internal corporate network. However, in the context of wireless sensor networks, the assumption about the physical security of the network infrastructure is unrealistic. The wireless shared medium is exposed to outsiders and susceptible to a wide range of attacks such as: jamming of the physical layer, disruption of the medium access control layer, attacks against the routing protocols, targeted attacks on the transport protocols, or even attacks intended to disrupt specific applications.

Much vulnerability in network protocols is caused by the lack of integrity and authentication mechanisms, which allows an attacker to alter or fabricate packets.

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 705–714, 2007.

Significant research in securing wired or ad hoc wireless routing protocols focused on this aspect. And network survivability is an essential aspect of reliable communication services. Survivability consists not only of robustness against failures occurring due to natural faults, accidents, and unintentional operational errors or misconfigurations, but also failures that are induced by malicious adversaries, particularly in the context of military networks. Mobile wireless sensor networks provide ubiquitous computing and undeterred access to the Internet, but significantly challenge survivability, both because users are mobile and because the communication channels are accessible to anyone [2].

Survivability in WSNS has six challenges: (1) wireless nature of communication, (2) resource limitation on sensor nodes, (3) very large and dense WSNS, (4) lack of fixed infrastructure, (5) unknown network topology prior to deployment, (6) high risk of physical attacks to unattended sensors. Moreover, in some deployment scenarios sensor nodes need to operate under adversarial condition. Thus, sensor nodes have security attacks due to the broadcast nature of transmission and to adapt to their environments, and establish a robust network protocol to communicate with the control center. Currently the survivability research and development of WSNS focus on two aspects: providing authentication and integrity of messages [3]–[7], and modeling of energy-efficient protocol [8]–[10]. The former is accomplished by applying some secure key based schemes. The latter is obtained by prolonging the lifetime of the WSNS without considering the intrusion. Energy-efficient protocols, although needed in order to prevent the sensor nodes from exhaustion, do not provide protection against injection and impersonation intrusions.

Therefore, motivation of this paper is to develop a model to evaluate the tradeoffs between the cost of defense mechanisms for WSNSs and the resulting expected survivability after a network attack. And based on the model we present a way to enhance the survivability of a WSNS. By varying the level of defense in the simulation, we can examine how this expected survivability changes with the defense level. Since costs are assumed to increase with the strength of the defense system, we can also derive a cost/survivability curve that managers can use to decide on the appropriate level of security for the network.

2 Model Description

We develop models of the incidents-process, the response of the system, and its survivability and the notations are introduced as below.

2.1 Notation

(i, j) denotes the index for incident. We consider actual, unauthorized incidents only. i denotes the prior incident and j the subsequent (or current) one.

$P(j)$ denotes probability that an incident is of type j .

$\tau(i, j)$ denotes inter-incident times between incidents i and j .

a denotes arrival rate of incidents .

d denotes index for system design, d in design space $\{D\}$.

m denotes index for defense mechanism, d in design space $\{M\}$, and $\{D \times M\}$ represents configuration space.

T denotes transition probability matrix with elements $\{p(r,s)\}$, where $\{p(r,s)\}$ possibly being functions of i, j, d and m.

l denotes (victim) sites where l in space $\{L\}$.

$h(l)$ denotes index for incidents at individual site l.

$H(l)$ denotes total number of incidents at site l.

$t(h(l),l)$ denotes time of h-th incident site l, $t(h(l),l)=\sum_{k=1}^{h}\tau(k)$,where $\tau(k)=t(k)-t(k-1)$.

n denotes number of simultaneous attacking sites in an incident.

$g(n|v)$ denotes probability density function for n with parameter v.

2.2 Model for the Incidents Process

In order to forecast incidents, the process is modeled as a marked, stochastic point process, where the incidents are the events that occur at random points in time, and the event type is the mark associated with an incident. The mark is used to identify random quantities associated with the point it accompanies. Each occurrence time t_k of the k-th incident in a temporal point-process has a mark j_k associated with it, where j_k will have values in a specified space. The mark, or event type in the case, has to take into account the severity of the incident and the possibility of single, or multiple and simultaneous attacks. This is because the wireless sensor network is lack of fixed infrastructure and is placed in an unbounded environment. Therefore the mark space will be 2-dimensional, characterized by type (severity) and number-of-attackers. That is, it will be in the $\{J \times N\}$ space. Although this 2-D marked point process model is developed, no data on the distribution of the number of attackers per incident are available, so only a 1-D mark space with severity was used in the simulations.

A stochastic point process can generally be represented as $\{x(t)|t \in T\}$, that is, as a family of random variables indexed by a parameter t that takes values in a parameter set T called the index set of the process. In this case, t represents time, and since T is a subset of R, it is a continuous-parameter process. The stochastic point process $\{x(t)|t \in T\}$ is completely characterized statistically by the joint distribution function for the random variables $x(t_1)$, $x(t_2)$, $\cdots$, $x(t_k)$ known for any finite collections $\{t_1, t_2, \cdots, t_k\}$ and $X_1, X_2, \cdots, X_k$ where $t_i \in T$ and $X_i \in R$ for $i = 1, 2, \cdots, k$.

$$P_{x(t_1),x(t_2),\cdots,x(t_k)}(X_1, X_2, \cdots, X_k) = \Pr\left(x(t_1) \le X_1, x(t_2) \le X_2, \cdots, x(t_k) \le X_k\right)$$

With every point process, there is an associated counting process denoted by $\{N(t)|t>t_0\}$ which indicates the total number of points in the interval $\lfloor t_0)$ regardless of their marks.

The characteristic functional for a sequence of independent random variables is given by:

$$\Phi x(iv)=E\left[\exp\left\{i\int_{t_0}^{T} v'(t)\,dx(t)\right\}\right]$$

Where $\{v(t)|t_0 \le t \le T\}$ is an arbitrary vector-valued function, the prime denotes the transpose operation, and $i=\sqrt{-1}$ here. For the purposes of analysis, the probability density function of the "inter-incident times" (τ 's) is considered which is denoted by $f(t)$. That is $f(t) = \Pr\{t \le \tau \le t+dt\}$. And when the process is Poisson, the density function is given by $f(t) = a*e^{-at}$, where a is the rate of occurrence of incidents and the distribution function is given by $F(t) = 1-e^{-at}$.

In this paper, the hypothetical data is used to run the simulations. And there are a number of issues in estimating the model parameters for the incidents-process:

1. The functional forms for inter-event times $f(t)$ should be determined. Frequently this is assumed to be Poisson, but this has to be verified by examining the distributions observed from the data. It may be that some other distribution such as the Weibull, or a mixture of exponential distributions and so on.
2. Once the form for $f(t)$ is determined, its parameters will have to be estimated.
3. Next $P(j)$, or the probabilities of each incident type j will be estimated.
4. It is also important to test whether $f(t)$ depends on i or j, or both.
5. Similarly, the issue of stationary of the process will have to be investigated.

So future work may introduce biases in parameter estimates, and it is important to take note of this.

2.3 Model for the System

Next we need to characterize the systems designs under consideration and the potential defense mechanisms that may be employed within the WSNs. That is, we need to define the design/architecture space $\{D\}$ of the system, and the defense mechanism state space $\{M\}$. The combination of a system design and defense mechanism will be called the configuration space $\{D\times M\}$. The design could include distributed sub-systems with different defenses for the sub-systems.

The response prediction model will predict the transition of the system to a new state after an attack/incident has occurred, and will be a function of the incident type and the configuration, or $p(r,s) = p(r,s|j,d,m)$. Thus, given an incident-type j and initial system state r, the subsequent state s may be any one of the set $\{S\}$ of possible states that the system can be in, such as normal, under attack, compromised, recovered, or non-functional. The actual states may of course be different for different configurations. The transition matrix T will probabilistically map r to s given j, d, m. That is, each element of T is the probability of the system of design d and defense mechanism m going to another (possibly compromised) state when subjected to an incident of type j. In general, the incident type j will be a vector of severity level and number of attackers. But since data on the number of attackers were not available, j is taken to be severity only in the simulations conducted here.

Without any loss of generality, the states of structure T are ordered by degree of compromise, that is, from $s=1$ to $s=S$ non-functional. Given an incident, the system can never go to a "better" state: therefore the lower triangle below the diagonals will have structural zeros as shown below.

The following constraints are imposed on the elements of T, $\{p(r,s)\}$ in terms of their dependence on s, j, m.

$p(r,s) \downarrow s(\forall s>r)$, holding j, m constant, that is, same severity level and same defense; this implies graceful degradation: the probability of going to a much worse state is lower than going to a slightly worse state.

$p(r,s) \uparrow r(\forall s>r)$, holding j, m constant, that is, same severity level and same defense; vulnerability increases with level of degradation.

Assuming that the j are ordered from most severe to least severe,

$p(1,1) \uparrow j$, holding m constant, that is, same defense level; probability of staying normal is higher if the incident is less severe.

$p(1,s) \downarrow j(\forall s>1)$, holding m constant, that is, same defense level; probability of degradation is lower if the incident is less severe.

$p(r,s) \downarrow m(\forall s>r)$, holding j constant, that is, same severity level; probability of degradation is lower if the defense is stronger.

$p(r,r) \uparrow m(\forall r)$, holding j constant, that is, same severity level; probability of staying in the same state and not degrading is higher if the defense is stronger.

$p(r,s) \uparrow n(\forall s>r)$, holding all else constant. Probability of degradation increases with the number of attackers.

And $\sum_s p(r,s) = 1$. This implies that the system must end up in some state or other.

If the transition probabilities in each case are known, the relative data is inputted directly into the model. Otherwise, a model is developed to generate the elements $\{p(r,s)\}$ of the transition probability matrix T, or compute them by considering the path through intermediate states during the attack-response episodes that the system may experience. Estimating these transition matrices is critical but extremely complex, since $S^2 \times J \times D \times M$ probabilities must be estimated. Thus some simplifying rules may be employed to generate them. Since data on these probabilities were not available, we considered only one value of D; that is, only one design in the simulation runs below. The model, however, has been designed to handle any number of designs as long as data on them are available.

Currently, no reliable data are available on the times to transition to different states, or the time to fully recover. Since it may be reasonably expected that recovery times will be shorter than that on the average, in these simulations we have assumed that the system would always fully recover before the next incident occurred. So the initial state r was always set equal to 1. However, the model includes the possibility of the system still being in a compromised state when the next incident occurs.

In the absence of data, a model is developed to generate the $p(1,x)$, such that

$$p(1,s) = p\left(s, j, \cos t(m); \pi_0, \chi_0, \pi_1, \chi_1, \pi_2, \chi_2\right)$$

There are two cases, $s=1$ and $s>1$.

$$p(1,1) = \pi_2 * \left(1 - e^{-\pi_1(\cos t(m) - \pi_o)}\right)(s=1)$$

$$p(1,s) = \chi_2 * \left(1 - e^{-\chi_1(\cos t(m) - \chi_o)}\right)(s>1)$$

These are simple but commonly used functional forms that are concave and convex respectively, and so reflect decreasing returns with cost. π_1 And χ_1 are the critical shape coefficients that determine the relationship of the transition probabilities with the cost of the defense mechanisms $\cos t(m)$. This in turn determines how the survivability varies with cost.

$\pi_2 = \pi_2(j)$ Which is modeled as a linear function = $\pi_3(j)$, and $\chi_2 = \chi_2(j,s) = \chi_3 * ((6-s) - (0.4 * j))$, again linear in s and j.

The scale coefficients π_3 and χ_3, as well as the constants, are calibrated to give reasonable values of the transition probabilities subject to all the restrictions given above. The location coefficients π_0 and χ_0 were set to 0, π_1 , χ_1 , π_3 and χ_3 are varied during the simulation runs.

In the future work, with data available, it may be possible to cluster the incident types and/or the system configurations into smaller subsets, since our interest is with respect to their impact on survivability only. This will make is easier for managers to assess the survivability of their systems.

3 Survivability Modeling

As discussed in the introduction, survivability is the key issue we wish to investigate with the simulation model. Therefore it is necessary to develop a measurable concept of survivability. There has been considerable work done on survivability in traditional computer network, and although that analysis is essentially at the network topology level.

Survivability is the degree to which a system has been able to withstand an attack or attacks, and is still able to function at a certain level in its new state after the attack. This new state s, in general will be a compromised state, and is the state in which the system ends up before any full-fledged recovery or repairs are done to it to restore it to its normal state. At the conceptual level, we propose that survivability be measured as:

SURV = (performance level at new state s) / (normal performance level)

The main issue is the measurement of performance levels. In telecommunications, it is generally taken as the traffic that is still carried relative to the offered traffic the network could carry under normal conditions. An analogous approach could be taken for computer systems, in that the different functionalities and services could be considered separately, and an assessment could be made as to what extent each functionality has survived in the new system state after an attack. For example, if a given functionality has survived intact, its value would be 1, and if the system were completely nonfunctional with respect to that service, then its value would be 0. Intermediate states would have values in between.

Let $\varphi(s,k)$ be the degree to which the compromised function/service k has survived in state s, and let $w(k)$ be the importance level of function/service. Then one possible measure of survivability might be in the form of a weighted sum:

$$SURV(s)=\sum_{k} w(k)*\varphi(s,k)$$

This assumes that a complete set of states $\{S\}$ of the system has been defined, and that a systems analyst can assess $\varphi(s,k)$ for each s and k . In view of the data requirements, it may be necessary to aggregate the state space $\{S\}$, and the different functionalities and services $\{K\}$. The states in $\{S\}$ may be {normal, under attack, compromised, recovered, non-functional}, for example, or {normal, minor compromise, significant compromise, very serious compromise, nonfunctional}. Then $\varphi(s,k)$ could be the average level to which function or service k survives in each of those states s . This is a flexible approach, and can be applied in many situations. For example, there might be a particular function that an organization values very highly (such as protecting the confidentiality of a database in a financial services company). Then the weight on this would be very high and also the survivability of this function could be rated low even for a slight compromise. Then any defense mechanism that protected this function would give a high expected survivability, and thus a high benefit, while a defense that did not protect this function would give very low value for expected survivability, and thus very low benefits.

This is a standard multi-criteria approach to assessing survivability. While this approach has been used widely, there can be difficulties and biases associated with such a measure. These can be mostly overcome through careful analysis. The weights $w(k)$ are such that $0\le w(k)\le 1$ and $\sum_{k} w(k)=1$;

The $\varphi(s,k)$ may also be normalized measures $0\le\varphi(s,k)\le 1$. Then $\mathrm{SURV}(\mathrm{s})$ will be between 0 and 1, where 0 means total failure and 1 means completely normal.

Another measure may be a “relative” survivability measure. To derive this, we consider the maximum level of functionality k in the normal state $X(k)$. However, the level of functionality required might be $x(k)$ where $0\le x(k)\le X(k)$. Let the level of functionality k that has survived in state s be $x'(k,s)$. Then we can define survivability relative to the requirement as

$\varphi'(k,s)=x'(k,s)/x(k)(x'<x)$ That is, the fraction of the required level that is available, and

$\varphi'(k,s)=1(x'>x)$, since the surviving level is greater than what is required.

Instead of a weighted survivability, we may consider the worst degree of compromise that has occurred across all functions and services. This would be analogous to "worst-case" survivability that is often considered in telecommunications. In that case,

$$SURV(s)=\min_k \varphi(k,s)$$

In many real situations one cannot be always aware of possible vulnerability of a system. However, it may be possible to enumerate the set of all possible compromises that could occur given the existence of (unknown or known) vulnerabilities. In such cases, we may proceed as follows:

Let the probability that function/service k is compromised to degree x by incident-type j be given by $p_{k,j}(x)$. Then we can simulate the overall compromise across all k, and compute the survivability after each incident, or we can simplify the analysis and consider the expected compromise $E\left[x(k,j)\right]$ given j,

Where $E\left[x(k,j)\right]=\int_0^1 x^* p_{k,j}(x)^* dx(0\le x\le 1)$ and compute survivability as

$$SURV\Big|j=\sum_k w(k)^*\left(1-E\left[x(k,j)\right]\right)$$

It may be desirable that the weights $w(k)$ reflect the utilization of a function or service in addition to its importance. On the other hand, we still need to distinguish the "essential" functions and services regardless of how much they are utilized. In such situations, we might partition the set $\{K\}$ into say, $\{K_0,K_1,K_2\}$, where K_0 is a set of unimportant functions/services, K_1 is a set of functions/services that are used/needed very often, but not critical, and K_2 is a set of essential functions/services.

Where k is in $\{K_1\}$ and k' is in $\{K_2\}$. The multiplicative term in both cases ensures that if an essential function/service fails, that is $\varphi(s,k')=0$ for any k', survivability goes to 0. The second form for $SURV(s)$ ensures that it does not always go to 0 when all the functions/services in set $\{K_2\}$ fail totally but some or all the functions/services in set $\{K_1\}$ have survived.

Another approach is to consider the relative degree of survivability of different systems and their configurations. It may be possible to conduct pair-wise comparisons of systems and sites and assess relative survivability of one site with respect to the other. If sufficient data can be obtained on some sites, then our proposed simulation model can be run to explore a variety of potential attack scenarios. Yet another approach to assessing survivability is to use "conjoint analysis" which is a method of evaluating the aggregate value of a product or service (in this case survivability) based on "part-worse" of the different feature or aspects that comprise survivability.

There is no "absolute" survivability and what we are interested in is assessing the strength of a current defense mechanism of a system of a given design relative to a stochastic incidents process. The actual survivability could be a function of many other factors such as the policies of the system managers, the "behavior" of the system and the deterrence it can induce among potential attackers, its reaction (detection, resistance, recovery), or the publicity surrounding an incident experienced by the WSNs.

4 Simulation

Our interest is to observe how well a system survives when subjected to a series of attacks. This will obviously depend on both the severity levels of the attacks as well as the level of defense that is built into the system. The stronger the defense system, the more likely it is to withstand an attack, that is to stay in its normal state, and less likely to end up in a compromised state. In other words, the transition probabilities of the system are a function of the defense mechanism, and this functional relationship drives the expected survivability of the system in any attack scenario. Therefore simulation was carried out for different probabilities of the attack types, and different relationships between the cost of the defense mechanism and the probabilities of the system ending in the various possible states.

A large number of simulations can be carried out with this model to investigate a wide variety of issues related to managing survivability. The Poisson process can be considered easily, because it is a flexible model and commonly used in point processes. However, any other distribution can be used instead, and the impact of alternative distributions such as the mixed exponential can also be investigated.

When survivability is not critical, the WSNs may choose a lower point on the trade-off curve, but when survivability is critical, the WSNs may well choose a point higher up on the curve. However, even if optimality is not aimed at, the WSNs can still use the curve to find the most appropriate point in the tradeoff between cost and survivability.

5 Summary

In this paper a model is developed that simulates complete episodes of attacks on network computer systems and the responses of these systems. This approach has involved developing a flexible template that can be used to analyze a variety of scenarios related to attacks on and survivability of network systems. The model encompasses several detailed aspects of the attack incidents, such as the type of attack, the number of attackers, and possible correlation between the rate of incidents and the type of incidents.

It can easily be extended to include trends or alternative distribution of inter-incident times. The system response has been modeled probabilistically through a state transition matrix where the transition probabilities are functions of the type of incident and the defense mechanism. A set of reasonable constraints on the probabilities and a model to generate them in the absence of data is outlined in this paper. The model reflects a relationship between the transition probabilities and the cost of the defense mechanism the system may have. The model can also be calibrated by expert opinion.

References

[1] Tilak, S., Abu-Ghazaleh, N.B., Heinzelman, W.: A taxonomy of wireless micro-sensor network models [J]. Mobile Computing and Communications Review 1(2), 1–8 (2002)

[2] Akyildiz, L.F., Su, W.L., Sankarasubramaniam, Y., Cayirci, E.: A survey on sensor networks [J]. IEEE Communications Magazine 40(8), 102–114 (2002)

[3] Carbunar, B., Ioannidis, I., N-Rotaru, C.: Janus: Towards robust and malicious resilient routing in hybrid wireless networks. ACM Workshop on Wireless Security (October 2004)

[4] Karlof, C., Wagner, D.: Secure routing in wireless sensor networks: Attacks and countermeasures. Ad Hoc Networks, pp. 293–315 (January 2003)

[5] Khalil, I., Bagchi, S., N-Totaru, C.: Dicas: Detection, diagnosis and isolation of control attacks in sensor networks. In: IEEE Conference on Security and Privacy for Emerging Areas in Communication Networks (SecureComm) 2005 (September 2005)

[6] Perrig, D.T.A., Canetti, R., Song, D.: The tesla broadcast authentication protocol. RSA CryptoBytes 5(2), 2–13 (2002)

[7] Perrig, A., Szewczyk, R., Wen, V., Culler, D., Tygar, J. (eds.): SPINS:Security Protocols for Sensor Networks. Proc, of ACM Mobicom 2001 (2001)

[8] Chen, B., Jamieson, K., Balakrishnan, H., Morris, R.: Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. In: Proceedings of the ACM International Conference on Mobile Computing and Networking, ACM, Rome, Italy (2001)

[9] Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H.: Energy efficient communication protocols for wireless microsensor networks. In: Proceedings of the Hawaii International Conference on Systems Sciences (January 2000)

[10] Wang, A., Chandrakasan, A.: Energy-efficient dsps for wireless sensor networks. In: ICASSP (2001)

Reducing End-to-End Delay in Multi-path Routing Algorithms for Mobile Ad Hoc Networks

Nastooh Taheri Javan and Mehdi Dehghan

Ad Hoc Network Labratoary, Computer Engineering Department,
Amirkabir University of Technology, Tehran, Iran
{nastooh, dehghan}@aut.ac.ir

Abstract. Some of the routing algorithms in mobile ad hoc networks use multiple paths simultaneously. These algorithms can attempt to find node-disjoint paths to achieve higher fault tolerance capability. By using node-disjoint paths, it is expected that the end-to-end delay in each path should be independent of each other. However, because of natural properties of wireless media and medium access mechanisms in ad hoc networks, the end-to-end delay between any source and destination depends on the pattern of communication in the neighborhood region. In this case some of the intermediate nodes should be silent to reverence their neighbors and this matter increases the average of end-to-end delay. To avoid this problem, multi-path routing algorithms can use zone-disjoint paths instead of node-disjoint paths. Two routes with no pair of neighbor nodes are called zone-disjoint paths. In this paper we propose a new multi-path routing algorithm that selects zone-disjoint paths, using omni-directional antenna. We evaluate our algorithm in several different scenarios. The simulation results show that the proposed approach is very effective in decreasing delay and packet loss.

Keywords: MANET; Routing Algorithms; Multi-Path Routing; Zone-Disjoint Paths.

1 Introduction

Mobile Ad hoc Networks (MANETs) are characterized by dynamic topology, high node mobility, low channel bandwidth and limited battery power. To provide end-to-end communication throughout the network, each mobile node acts as an intermediate router forwarding messages received by other nodes.

Designing efficient routing protocols is the central challenge in such dynamic wireless networks. However, many ad hoc routing algorithms have been proposed, such as AODV [1], DSR [4]. Routing protocols for MANETs can be broadly classified into reactive (on-demand) and proactive algorithms [1]. In reactive protocols, nodes build and maintain routes as they are needed but proactive routing algorithms usually constantly update routing table among nodes.

In on-demand protocols, nodes only compute routes when they are needed. Therefore, on-demand protocols are suitable for dynamic large networks. When a

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 715–724, 2007.

node needs a route to another node, it initiates a route discovery process to find a route. On-demand protocols consist of the following two main phases.

Route discovery is the process of finding a route between two nodes. Route maintenance is the process of repairing a broken route or finding a new route in the presence of a route failure.

Among the on-demand protocols, multi-path protocols have relatively greater ability to reduce the route discovery frequency than single path protocols. On-demand multi-path protocols discover multiple paths between the source and the destination in a single route discovery. Therefore, a new route discovery is needed only when all these paths fail. In contrast, a single path protocol has to invoke new route discovery whenever the only path from the source to the destination fails. Therefore, on-demand multi-path protocols cause fewer interruptions to the application data traffic when routes fail. They also have lower control overhead because of fewer route discovery operations.

Multi-path Routing can provide some benefits, such as load balancing, fault-tolerance capability, and higher aggregation of available bandwidth. Load balancing can be achieved by spreading the traffic along multiple routes; this can alleviate congestion and bottlenecks. From fault tolerance perspective, multi-path routing can provide route resiliency. Since bandwidth may be limited in a wireless network, routing along a single path may not provide enough bandwidth for a connection. However, if multiple paths used simultaneously to route the traffic, the aggregation of the paths may satisfy the bandwidth requirement of the application and a lower end-to-end delay may be achieved. Moreover, the frequency of route discovery is much lower if a node maintains multiple paths to destination.

After recognizing several paths between the source and the destination in route discovery process in multi-path routing algorithms, data transferring can be started through several routes. By using these mechanisms we can distribute the traffic to several paths in order to balance the traffic, and increase the bandwidth and as a result decaling the delay.

Choosing the suitable paths to the destination for transferring data traffic is the most important issue. Choosing disjoint paths between the source and the destination is one of the ideas. This increases the fault tolerance noticeably. If there isn’t any common node in the paths chosen for transferring the data, it may break down and also the route may spoil.

As we know there are two problems in wireless networks, known as "hidden station" and "exposed station". For handling these problems, CSMA/CA [11] protocol has been suggested. In 802.11 standards, this protocol is used for accessing the channel. Due to transferring RTS and CTS packets between nodes in this protocol, some of the nodes do not transfer the data and as a result the delay is increased.

As an example, consider figure 1 that shows an imaginary LAN with ten nodes. In this figure radio range of every node is illustrated and the dotted line shows the relation between nodes. In other words, the dotted lines between any two nodes show that they are located in the radio range of each other.

There are two node-disjoint paths, S-I1-I2-I3-I4-D and S-I5-I6-I7-I8-D, between S and D, which transferring the data in one path is not completely separated from the other path. In this case, the delay of every path is related to the traffic of the other path, because of transferring RTS and CTS packets between the nodes of the network in order to avoid the collision and solve hidden station and exposed station problems. As a result some of the stations in a path in order to receive CTS from a node in the opposite path should postpone their sending.

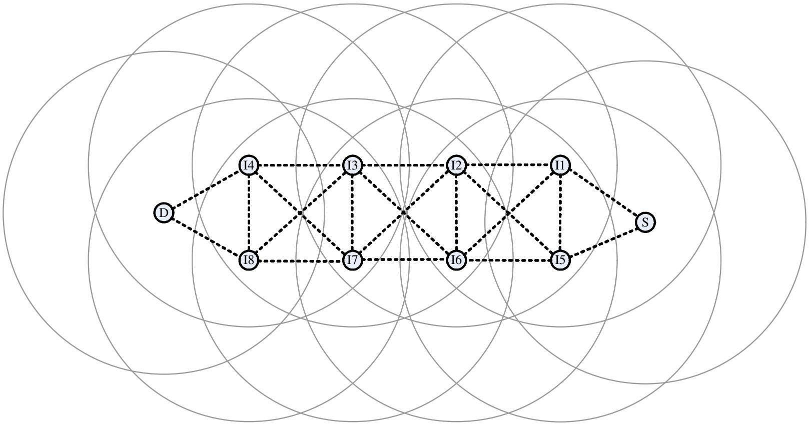


Fig. 1. Node-Disjoin paths

To solve this problem, we can use zone-disjoint paths instead of node-disjoint paths. Two routes with no pair of neighbor nodes are called zone-disjoint. In [7, 8], the authors proposed a method for distinguishing the zone-disjoint paths in the networks equipped with the directional antennas. However most of the present equipments are not equipped with directional antenna. In this paper a multi-path routing algorithm is proposed. In this approach, by using omni-directional antennas, the zone-disjoint paths are recognizable and these paths can be used for sending the data traffic simultaneously.

The rest of this paper is organized as follows. The following section deals with the related works. Section 3 describes the proposed protocol in detail. Performance evaluation by simulation is presented in section 4, and the concluding remarks are made in section 5.

2 Related Works

Multi-path routing and its applications have been well studied in wireless ad hoc networks.

The goal of SMR [6] is finding maximally disjoint multiple paths. SMR is an on-demand multi-path source routing algorithm that is similar to DSR [4]. To discovery the routing paths, the source, at first, broadcasts the RREQ to every neighbor. When the RREQ is delivered to a node, the intermediate node's ID is included into packet. Then the node, receiving RREQ, re-broadcasts it to every outgoing path. In this algorithm, the destination sends a RREP for the first RREQ it receives, which represents the shortest delay path. The destination then waits to receive more RREQs. From the received RREQs, the path that is maximally disjoint from the shortest path is selected and the destination sends a RREP for the selected RREQ. In SMR, the intermediate nodes do not reply to RREQs, this is to allow the destination to receive RREQs from all of the routes, so that it can select the maximally disjoint paths.

AOMDV [3] is an extension to AODV [1] protocol for computing multiple loop-free and link-disjoint paths. In AOMDV through a modified route discovery process multiple link-disjoint paths are computed. The destination responds to only those unique neighbors from which it receives a route request. Each node in the network maintains a list of alternate next hops that are stored based on the hop count. If during routing, one of the links between any two nodes breaks, then the immediate upstream node switches to the next node in its list of next hops. In this algorithm, the source node initiates a route request when all of its alternate paths fail. The main drawback of this protocol is that the alternate paths that are computed during route discovery are not maintained during the course of data transfer.

Multi-path Source Routing (MSR) [9] is an extension of DSR [4] protocol. It consists of a scheme to distribute traffic among multiple routes in a network. MSR uses the same route discovery process as DSR with the exception that multiple paths can be returned, instead of only one (as with DSR). When a source requires a route to a destination but no route is known (in the cache), it will initiate a route discovery by flooding a RREQ packet throughout the network. A route record will be contained in header of each RREQ in which the sequence of hops that the packet passes through is recorded. An intermediate node contributes to the route discovery by appending its own address to the route record. Once the RREQ reaches the destination, a RREP will reverse the route in the route record of the RREQ and traverse back through this route. Each route is given a unique index and stored in the cache, so it is easy to pick multiple paths from there. Independence of the paths is very important in multi-path routing. Therefore, disjoint paths are preferred in MSR. As MSR uses the same route discovery process as DSR, where the complete routes are in the packet headers, looping will not occur. When a loop is detected it will be immediately eliminated.

Since source routing approach is used in MSR, intermediate nodes do nothing but forward the packets according to the route indicated in the packet-header. The routes are all calculated at the source. A multiple-path table is used for the information of each different route to a destination. This table contains the following items for each route to the destination: the index of the path in the route cache, the destination ID, the delay (based on estimated RTT), and the calculated load distribution weight of a route. The traffic to a destination is distributed among multiple routes; the weight of a route simply represents the number of packets sent consecutively on that path.

In [7, 8] multi-path routing with directional antenna is proposed. In this protocol directional antenna is used for finding zone-disjoint paths between a source and a destination. Due to low transmission zone of directional antenna, it is easier to get two physically close paths that may not interfere with each other during communication.

3 The Proposed Algorithm

The proposed algorithm can be used in all on-demand routing protocols. In on-demand protocols when the source has data packets to send but does not have the route information to the destination, it floods the RREQ packet to search and discover a route to the destination.

We can say generally the destination in the proposed algorithm tries to choose the zone-disjoint paths from received RREQs and send the RREPs to the source for these RREQs. For recognizing zone-disjoint paths between the source and the destination, a new field is added in RREQ packet, which is called *ActiveNeighborCount* and it is initialized to zero. As a matter of fact this field shows the number of active neighbors for the nodes on a path. Active neighbor is a node received this RREQ, and the source and the destination may choose another path which has this node on it, and in this case sending the data from selected paths, is related to each other. In order to set the proposed algorithm working, the entire nodes should keep a table which is called RREQ_Seen. This table records the characteristics of received RREQs by every node.

Finally for the last important change in on-demand algorithms, the intermediate node should not send RREP to any source and in fact should let the destination receive all RREQs and choose the best paths and send RREPs to the source. In other words, in the proposed algorithm, the intermediate nodes do not need using Route Cache.

In this algorithm, like other on-demand algorithms, the source node floods a RREQ packet in order to recognize a route to the destination. As mentioned before, initial value of *ActiveNeighborCount* in this packet is zero. In this case every intermediate nodes which received the RREQ, records it's characteristics in RREQ_Seen table, but before sending this packet, asks its neighbors "Have you ever seen this RREQ with this characteristics before?" and sends a packet which is called RREQ_Query to its neighbors and waits for their reply for a specified time distinguished by a timer. In this case the neighbors have to reply the answer by searching in their RREQ_Seen table. When the time is over, this node increases the value of *ActiveNeighborCount* in RREQ packet with the number of neighbors that send positive answer, and then it floods RREQ packet to its neighbors.

In this case when the destination receives all of RREQs, it starts to choose disjoint paths and then between the chosen paths considers the values of *ActiveNeighborCount*s and chooses the paths which have less values of *ActiveNeighborCount*. In fact the destination by choosing the paths which have less values of *ActiveNeighborCount*, tries to select the zone-disjoint paths. Then the destination sends the RREP packets to the source through the chosen paths. As soon as the source receives the first RREP, it starts to transfer the data by this route and after receiving the next RREP, it divides the traffic into the present routes based on load balancing criteria.

To clarify the strategy of the proposed algorithm, consider the given network of figure 2. Suppose in this case the node S is going to send data to the node D and it intends to send these data through two routes simultaneously. By paying attention carefully at the figure, you will find out there are three node-disjoint paths between S and D: S-A-D, S-B-D, and S-C-D. Now if the source chooses S-A-D and S-B-D paths as an example, due to transferring RTS and CTS between A and B, we can say one of the nodes can be active in a given time. Although the data is transferring from two routes, but it seems that just one path is active in each time.

Now suppose the proposed algorithm has been used. On the first step, the source sends RREQ to its neighbors, i.e. A, B and C. However, these nodes before sending this packet to their neighbors should ask them about this RREQ. After doing this, the A and C nodes find out just one of their neighbors has seen this RREQ before. Therefore A and C add a unit to the *ActiveNeighborCount* field of their RREQ. However, B node finds out two of its neighbors have seen this RREQ before and adds two units to the *ActiveNeighborCount* field of its RREQ. Then these nodes send their RREQ to their neighbors.

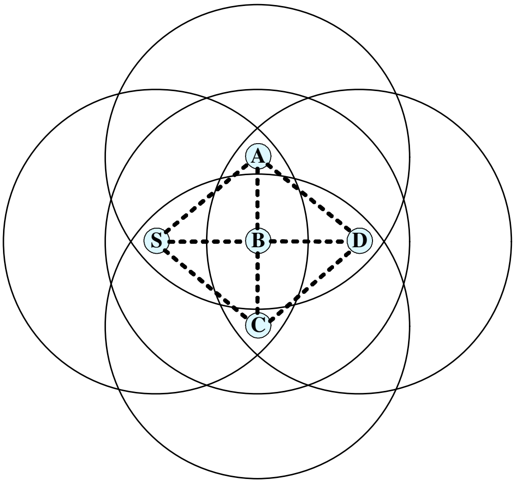


Fig. 2. Zone-Disjoint paths

Finally the destination receives several RREQs and finds out three paths between A and D: S-A-D, S-B-D and S-C-D which are node-disjoint paths. Then the destination considers the *ActiveNeighborCount* field in these RREQs and chooses two of them: S-A-D and S-C-D routes as the best paths and sends the RREP packets to the source by these two routes which are zone-disjoint.

In the existing algorithms, they just considered the node-disjoint routes and did not consider the negative effect of the neighbor routes in the performance of packet forwarding. However, by using the proposed idea the neighbor routes have the minimum effect on each other. It is very important to remember that the proposed idea just uses the omni-directional antenna in order to choose the zone-disjoint routes to send the data.

4 Performance Evaluation

In order to demonstrate the effectiveness of the proposed algorithm, we evaluated it and compare its performance to SMR. We implemented the proposed algorithm in DSR routing protocol, from now on we name this protocol as Proposed-DSR.

4.1 Simulation Environment

As simulation environment we use GloMoSim [10]. The simulated network is deployed in a flat square with 1000 meters on each side. The network was modeled with mobile nodes placed randomly and all nodes have the same transmission range of 250 meters. The radio model to transmit and receive packets is RADIO-ACCNOISE which is the standard radio model used. The IEEE 802.11 was used as the medium access control protocol. The random waypoint model was adopted as the mobility model. In the random waypoint model, a node randomly selects a destination from the physical terrain. It moves in the direction of the destination in a speed uniformly chosen between a minimum and maximum speed specified. After it reaches its destination, the node stays there for a time period specified as the pause time. In our simulation, minimum speed was set constant to zero. All data packets are 512 bytes and each simulation time is 300 seconds.

4.2 Performance Metrics

Three important performance metrics were evaluated: (i) The average of end-to-end delay of data packets – this includes all possible delays caused by buffering during route discovery latency, queuing at the interface queue, retransmission at the MAC, propagation and transfer times; (ii) Packet delivery ratio; (iii) Control overhead ratio - Ratio of the number of routing control packets to the total received packets.

4.3 Simulation Results

In the first scenario, to evaluate the capability of the protocols in different node mobility, we change node mobility by varying the maximum speed. The number of nodes and pause time was fixed at 100 nodes and 1 second, respectively.

In this scenario the proposed-DSR exhibits the lower end-to-end delay than the SMR (Fig. 3), and it also has greater packet delivery ratio than SMR (Fig. 4). However, in this case the proposed-DSR has greater control overhead ratio than SMR (Fig. 5).

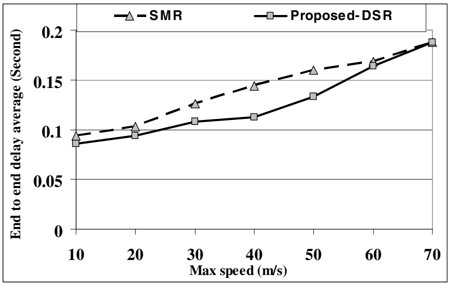


Fig. 3. The speed of the nodes vs. the average of end-to-end delay

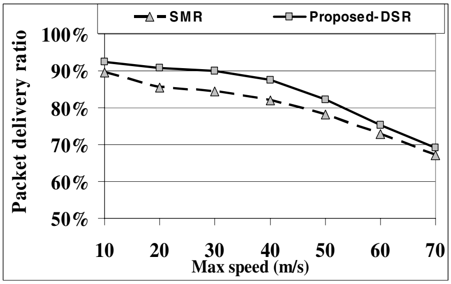


Fig. 4. The speed of the nodes vs. packet delivery ratio

In the second scenario, the effect of node mobility with different velocities on the performance of the routing protocols is evaluated. To achieve this, we change node mobility by varying the pause time. The number of nodes was fixed at 100 and the maximum speed was fixed at 25 m/s.

In this case, the proposed-DSR has lower end-to-end delay than SMR (Fig. 6), and it also exhibits greater packet delivery ratio than SMR (Fig. 7). The experimental results represent that if the pause time of the nodes increases, the amount of receiving data packets by destination nodes will be increase too. In this scenario, the proposed-DSR has greater control overhead than SMR (Fig. 8).

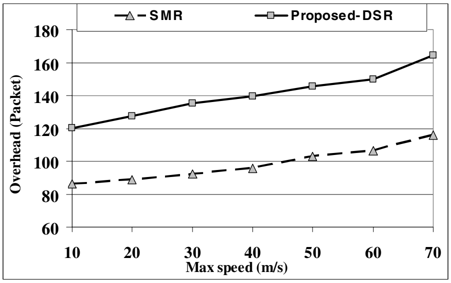


Fig. 5. The speed of the nodes vs. overhead

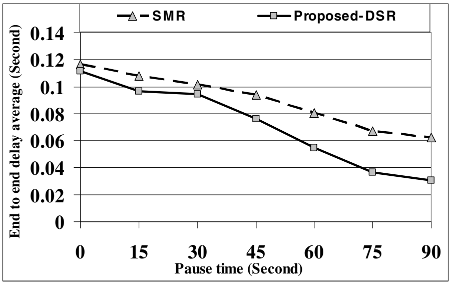


Fig. 6. Pause time vs. the average of end-to-end delay

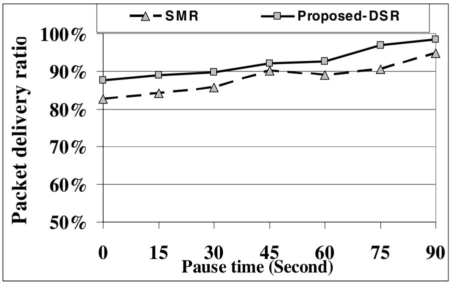


Fig. 7. Pause time vs. packet delivery ratio

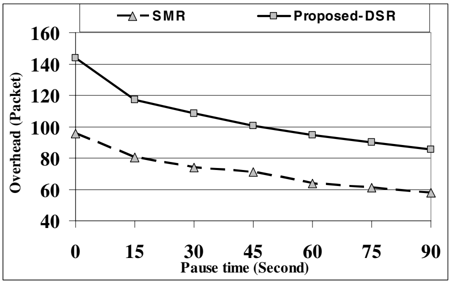


Fig. 8. Pause time vs. overhead

In the last scenario, we evaluate the proposed protocol by examining the effect of the density of nodes (the number of nodes) on the performance of proposed-DSR, and SMR protocols. To achieve this, we consider different number of nodes and in this case, the maximum speed and pause time were fixed at 25 m/s and 1 second, respectively.

In this scenario the proposed-DSR exhibits lower end-to-end delay than the SMR (Fig. 9), and it also has greater packet delivery ratio than SMR (Fig.10). Finally in this case the proposed-DSR has greater control overhead ratio than SMR (Fig. 11).

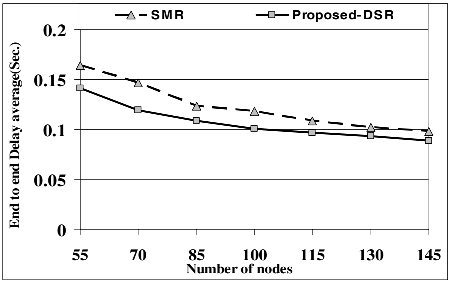


Fig. 9. The number of nodes vs. the average of end-to-end delay

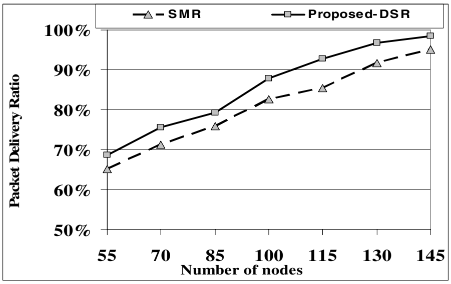


Fig. 10. The number of nodes vs. packet delivery ratio

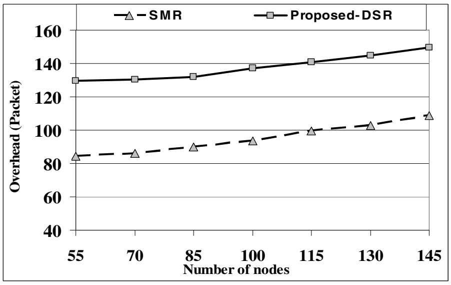


Fig. 11. The number of nodes vs. overhead

5 Conclusion

In some of the multi-path routing algorithms in MANETs, the source node spreads the data traffic to the destination node through several routes simultaneously. It seems that in these scenarios, the node-disjoint routes are the best option. However, the node-disjoint routes are not independent of each other and due to nature of MANET MAC Protocols (e.g. CSMA/CA), sending data by a route affects on the other routes. In this paper, a new multi-path routing algorithm is proposed in which by using common and omni-directional antennas we can recognize the zone-disjoint routes between any two nodes and use these routes for sending the data traffic. The simulation results show that the proposed algorithm is very effective in decreasing the packet loss ratio and also decreasing the average of end-to-end delay in MANETs.

Acknowledgments. This work was supported by Iran Telecommunication Research Center (ITRC).

References

1. Royer, E., Toh, C.: A Review of Current Routing Protocols for Ad-hoc Mobile Wireless Networks. IEEE Personal Communication Magazines, 46–55 (1999)
2. Sesay, S., Yang, Z., He, J.: A Survey on Mobile Ad Hoc Wireless Network. Information Technology Journal 2, 168–175 (2004)

3. Marina, M.K., Das, S.R.: On Demand Multipath Distance Vector Routing in Ad hoc Networks. In: IEEE International Conference on Network Protocols (ICNP), pp. 14–23, California, USA (2001)
4. Johnson, B., Maltz, D.A.: Dynamic Source Routing in Ad-Hoc Wireless Networks. Mobile Computing 353, 153–181 (1996)
5. Sambasivam, P., Murthy, A., Belding-Royer, E.M.: Dynamically Adaptive Multiparh Routing based on AODV. IEEE Communications Magazine 1, 205–217 (2002)
6. Lee, S.J., Gerla, M.: Split Multipath Routing with Maximally Disjoint Paths in Ad hoc Networks. In: IEEE International Conference on Communication (ICC), Helsinki, Finland, pp. 3201–3205 (2001)
7. Roy, S., Saha, D., Bandyopadhyay, S., Ueda, T., Tanaka, S.: Improving End-to-End Delay through Load Balancing with Multipath Routing in Ad Hoc Wireless Networks using Directional Antenna. In: IWDC 2003. LNCS, vol. 2918, pp. 225–234. Springer, Heidelberg (2003)
8. Bandyopadhyay, S., Roy, S., Ueda, T., Hasuike, K.: Multipath Routing in Ad hoc Wireless Networks with Directional Antenna. Personal Wireless Communication 234, 45–52 (2002)
9. Wang, L., Shu, Y., Dong, M., Zhang, L., Yang, O.W.W.: Adaptive Multipath Source Routing in Ad hoc Networks. In: IEEE International Conference on Communications (ICC), Helsinki, Finland, pp. 867–871 (2001)
10. Bajaj, L., Takai, M., Ahuja, R., Bagrodia, R., Gerla, M.: Glomosim: a Scalable Network Simulation Environment. Technical Report 990027, Computer Science Department, UCLA (1999)
11. Colvin, A.: CSMA with Collision Avoidance. Computer Communication 6, 227–235 (1983)
12. Mueller, S., Tsang, R., Ghosal, D.: Multipath Routing in Mobile Ad Hoc Networks: Issues and Challenges. In: Calzarossa, M.C., Gelenbe, E. (eds.) Performance Tools and Applications to Networked Systems. LNCS, vol. 2965, pp. 209–234. Springer, Heidelberg (2004)
13. Bose, P., Morin, P., Stojmenović, I., Urrutia, J.: Routing with Guaranteed Delivery in Ad Hoc Wireless Networks. Wireless Networks 7, 609–616 (2001)

Secure Intermediary Caching in Mobile Wireless Networks Using Asymmetric Cipher Sequences Based Encryption

Ahmed Reda Kaced and Jean-Claude Moissinac

GET-Télécom Paris, CNRS - UMR 5141,
37-39 Rue Dareau, 75014 Paris, France
{kaced,moissinac}@enst.fr

Abstract. The use of proxies is commonplace in today's MANETs, where they are used for a huge variety of network services. However the main problem of using proxies is that the end-to-end nature of the communication is broken. This leads to some severe security problems. One of the main questions that arise is how content caching by intermediaries can be done when end-to-end security is required.

In this paper, we will address the research issues of when and how end-to-end security, like confidentiality and authenticity can be preserved, in mobile ad hoc network communications, when having one or more cache proxies and router nodes in the data path.

We propose a solution for an encryption scheme based on Asymmetric Cipher Sequence which allows to an intermediary nodes to cache data and convert the ciphertext for one person into the ciphertext for another person without revealing the secret decryption keys or the plaintext.

Implementation results shows that we can simultaneously achieve high encryption through a wireless links.

1 Introduction

Security issues in Mobile Ad hoc NETworks (MANETs) [12] have drawn the attention of the research community in the last few years, driven by a wealth of theoretical and practical challenges. This growing interest can be largely attributed to new applications enabled by large-scale networks of small devices capable of performing complex tasks traditionally reserved for the more powerful terminals. Significant results in this area over the last few years have ushered in a surge of civil and military applications. As of today, most deployed MANETs are inter-connected with the wired networks and share the same applications. Except that, in the MANETs part, most of these applications have low bandwidth demands, and are usually delay tolerant.

Equally important, the consumer market is seeing a dramatic shift in usage habits, with cellular and other portable networked devices increasingly being used for a new range of audio and video-based entertainment/media applications, beyond simple voice communication. This can be seen in the slew of newly launched or highly anticipated services, such as VoIP, video streaming, video and

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 725–736, 2007.

audio downloads and playback, mobile TV and interactive games, specifically targeted to the wireless market.

To make these services scalable, one of the most common solutions found in the literature is the use of intermediary proxies [11] [8] [5]. A proxy is an intermediary placed in the path between a server and its clients. Proxies are used for saving network bandwidth, reducing access latency and coping with network and device heterogeneity.

However, one major problem with the proxy-based communication approach is the risk of revealing the original exchanged data to unauthorized parties. For example, when the original data are sent from a server to a proxy, anyone that eavesdrops on the communication link between the source and the proxy can gain access to the data information.

In this paper, address the issue of end-to-end security for different types of proxied MANETs, and present our approach for a proxy encryption framework having the following security properties:

– Proxies can cache encrypted data and decryption will only happen at the clients side. Therefore, the original data will not be revealed at any intermediate node.
– Proxies only perform encryption operations. This reduces the computational overhead at the proxy level, and hence allows one to build a more scalable proxy system.
– Data encryption and decryption operations are based on well accepted encryption theory that it is computationally infeasible to extract the original data.
– Member collusion are avoided, such that the intruder still cannot derive the original data, even if it have the decryption key of client, the encrypted data of client, and possibly all the encryption keys.

This paper is organized as follows. Section 2 presents some related works closed to our subject, when Section 3 present our approach for an encryption model based on the Cipher Sequence. We present also an algorithm to implement an Asymmetric Cipher Sequence to achieve the claimed security properties for a caching proxy. Section 4 presents an overview of the implementation of our proxy system architecture. We then report some results that illustrate the achievable encryption data rate using our technique, and give quantitative and qualitative analysis of the encrypted content delivery. Section 5 concludes and gives some possible future works.

2 Related Work

As outlined in [4], in a mobile environment, proxies can be used to decentralize the authentication process and allow the application to use a public-key security model on the wired network that may require a high computational effort, while keeping the computed functions at the device as simple as possible.

Recently, quite a few papers have been published to address the new security and privacy challenges in mobile computing [6] [3] [1] [13]. Among these several solutions, we find some proxy-based ones.

Burnside et al present in [2] a proxy-based architecture where proxies implement a public-key security model to control the access over shared resources (file, printer, etc.). For guaranteeing security and privacy this work uses two separate protocols: a protocol for secure device-to-proxy communication and a protocol for secure proxy-to-proxy communication.

The protocol for device-to-proxy communication sets up a secure channel that encrypts and authenticates all the messages in the wireless link using symmetric keys. The HMAC-MD5 algorithm is used for authentication and the RC5 algorithm for encryption. On the other hand, the proxy-to-proxy protocol uses the SPKI/SDSI(Simple Public Key Infrastructure/Simple Distributed Security Infrastructure) [7] to implement the access control through ACLs (Access Control Lists) on the public or protected resources.

Pearce et al use proxy in [10] in an application-layer SPKI/SDSI protocol which aims to provides secure communications, authentication and fast reauthentication over ad-hoc Wireless Sensor Networks.

In our solution, we consider the problem of secure proxy caching over a multihop ad hoc network. We exploit also proxies for content encryption to avoid any type of collusion attack. In our study we assume that the nodes int the MANET have a reasonable computation capability. Our main focus is on reducing the communication cost. We implement a set of protocols that use locality to reduce the communication complexity of secure data caching for dynamic groups of mobile multihop nodes. We aim at reducing the overall network communication cost using an anycast type of group join. Nodes attach to the network through the closest proxy already within the group.

3 Our Proposed Approach

In this section we describe our approach, the network architecture and the proposed protocols.

3.1 Proxy-Based Adaptation Delivery

In our architecture, a proxy is an active intermediary placed between two communication endpoints, typically a client and a server. Such an intermediary can be used to adapt a communication session to the type of clients and networks involved either by doing content encryption or network protocol enhancement. The proposed approach can easily be extended to a multilevel proxy structure. This is an important property since it is common to have multiple proxies along the communication path between the sender and a receiver.

The major advantages of using a proxy-based architecture for serving network clients, when compared to an end-to-end approach, are the following:

- all mobility - and wireless-dependent transformations (translation, transcoding) can be assigned to the proxy and need not be handled by the servers, allowing legacy services to be directly used for mobile access;
- all processing required for protocol and content transformations is distributed to other nodes where they are required, avoiding an overload at the servers;
- placing proxies at (or close to) a node with the wireless interface enables more agile and accurate monitoring of the wireless link quality, detection of mobile devices disconnections, as well as better selection of the required adaptation;
- transformations at any communication layer can be implemented, and are more easily adapted/customized according to the specific capabilities of the wireless links.

3.2 Security Services and Mechanisms

Our basic scheme for secure proxy caching on MANETs is based on maintaining a physical security tree of the members in the MANET. Our approach is based on the multicast security solution proposed in [9] and reused after in [14] which inspires our work, exploring the efficient use of Cipher Sequences on encryption operations, moreover we take into account MANETs constraints.

The basic idea is that a proxy given a proxy key, could convert the ciphertext for one person into the ciphertext for another person without revealing the secret decryption keys or the plaintext.

Solution theory. Molva and Pannetrat proposed in [9] the concept of **Cipher Sequence (CS)** for a security in multicast groups. The idea was as follows: by distributing secure functions to intermediate nodes, keying material has a dependency on the network topology. Therefore the containment of security exposures is assured.

Let $(k_{1 \leq i \leq n})$ be a finite sequence of n elements, where the elements are not necessarily unique, and let $\varphi : \mathbb{N}^2 \longmapsto \mathbb{N}$ be a function with the following

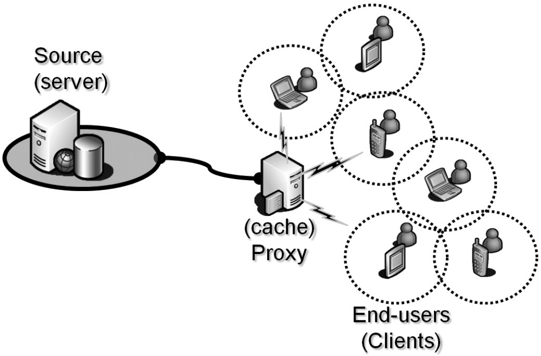


Fig. 1. Reference architecture for a proxy-based caching in MANETs

property: if $A = \varphi(B, k)$, it is computationally infeasible to compute k knowing A and B.

Assume S_0 is the information to be sent to different nodes. $S_{0 \leq i \leq n}$ is a finite sequence of $n + 1$ elements defined as:

$$\begin{array}{l} S_0, \text{ the initial value of the sequence.} \\ \text{for } i > 0 \;\; S_i = \varphi(S_{i-1}, k_i). \end{array}$$

Each node N_i is assigned a secret function φ_i. N_i receives data from its parent node N_j, computes $S_i = \varphi_i(S_j)$, and forwards the result S_i to its children. A leaf eventually receives S_n^i. Each leaf is given a reversing function $\bar{\varphi}_i$, and it can use $\bar{\varphi}_i$ to get the original data by calculating: $S_0 = \bar{\varphi}_i(S_n^i)$.

Such a sequence will be called *Cipher Sequence* associated to φ or CS_φ if, for all couples $(i, j) \in \mathbb{N}^2$ verifying $-1 \leq i \leq j \leq n$, there exists a computable function $\omega_{i,j}$ such as $S_i = \omega_{i,j}(S_j)$.

For example, Figure 2 depicts a simple tree with five cipher sequences. We follow one cipher sequence from the root (Server, S) to the leaf C_4 (Client 4). First, the root computes $\varphi_1(S_0)$ and sends the result to its children inner nodes. P_2 (Proxy 2) receives $S_1^4 = \varphi_1(S_0)$, computes and sends $\varphi_3(S_1^4)$ to R_3. Then R_3 receives $S_2^4 = \varphi_3(S_1^4)$ and sends $\varphi_6(S_2^4)$ to the leaf C_4. Finally, the leaf C_4 receives $S_3^4 = \varphi_6(S_2^4)$ and recovers the original data by computing $S_0 = \bar{\varphi}_4(S_3^4)$.

In MANETs, each node may acts as a router. Data sent from a server to an end-user may pass through some router nodes. In our architecture shown in Figure 1, data are cached in a proxy before being sent toward the end-users. In the example in Figure 2 and described before $\mathcal{S}$ is the server (source), $\mathcal{P}_i$ are caching proxies, $\mathcal{R}_i$ are router nodes and $\mathcal{C}_i$ are end users.

A CS is called *Symmetric Cipher Sequence* of φ denoted SCS_φ if the function $\omega_{i,j}$ can be computed from the knowledge of the sub-sequence $\{k_{i+1}, \ldots, k_j\}$.

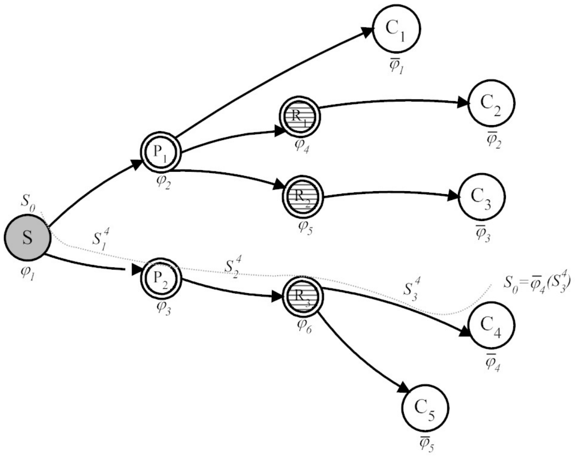


Fig. 2. An example of Cipher Sequence tree

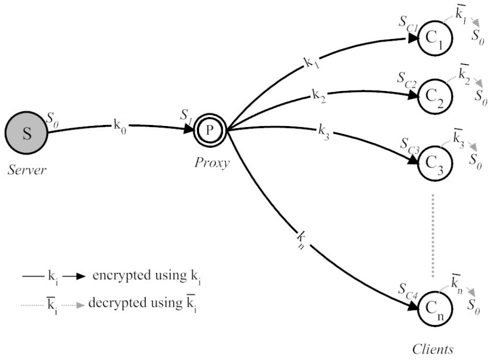


Fig. 3. ACS for secure caching proxy

A CS is called *Asymmetric Cipher Sequence* of φ denoted ACS_φ if it is computationally infeasible to determine the function $\omega_{i,j}$ based on the knowledge of the sub-sequence $\{k_{i+1}, \ldots, k_j\}$.

We use the properties of an ACS to implement a secure caching proxy illustrated in figure 3. The proxy can request S_1, the encrypted version of the original data S_0, from the source. Based on an encrypted key k_0, the source will transmit the encrypted data S_1 to the proxy, which will be cached at the proxy's local storage. When a client C_i requests the data, the proxy will further encrypt S_1 using the encryption key k_i and send the resulting encrypted data S_{Ci} to client C_i. C_i, upon receiving S_{Ci}, can decrypt the data to obtain the original data S_0, if C_i is given a decryption function $\omega_{0,i+1}$ (this is a property of CS). In addition, when the encryption is carried out using an ACS, then even when an entity holds on to all the encryption keys for k_i, it still cannot decrypt any of the encrypted data being cached, for $0 \le i \le n$, in order to obtain the original data S_0.

With this solution, our proxy cache can ensure the following security criteria for the transmitted data:

- confidentiality during transmission
- end-to-end confidentiality
- confidentiality against member collusion
- confidentiality against proxy intruders

Encryption model. Our encryption model is similar to that in the cipher sequences framework. We use Figure 2 again to describe our encryption model. In a MANET, nodes may be routers, form a multicast tree to transmit multicast traffic. The root $\mathcal{S}$ is the source, intermediate nodes $\mathcal{P}_i$, where i is an integer, are cache proxies, and every leaf node represents a set of local subgroup members attached to the same proxy. $\mathcal{R}_i$ are members of the MANET who act as router nodes.

$\mathcal{C}_i$ is the set of local members attached to $\mathcal{P}_i$. Each proxy may have local subgroup members and/or downstream proxies. Note in the cipher sequence framework, intermediate nodes do not have local members.

3.3 Protocol Description

In this section we give more details about our server-proxy-client MANET based architecture. In this architecture, a Client C_i request data from the server $\mathcal{S}$ passing via the proxy $\mathcal{P}$. We distinguish to possible scenarios according to requesting for a cached data or non cached yet. In the first scenario, data are not yet cached in the proxy level, so the server have to encrypt data and sends to the proxy, this later re-encrypts them and forwards to the client. In the second scenario, data are already encrypted and sent to the proxy which cache them in its local storage, the proxy have just to re-encrypt and sent these data to the client. In the following we describe the two possible scenarios.

Scenario 1: Not cached Data. Let us first consider the case wherein the data are not yet cached at the proxy. Figure 4 illustrates the operations between the server and the proxy, and the operations between the proxy and the client for requesting and caching the data. These operations are:

authentication. The client C_j sends a request to the server $\mathcal{S}$ to receive the content S_0 passing via the proxy $\mathcal{P}$, It joins its certificate of its identity Id_{C_i}. The proxy forward the request to the server $\mathcal{S}$. Lets the client c_i's certificate be $Id_{C_i} = (A_i, [A_i]_{B_i})$, where A_i and B_i are C_i's public and private keys, respectively, for RSA cryptography, and $\{S_k\}$ denotes information S encrypted with key k. The server keeps a record of the public key for each of its authenticated clients. To verify the identity of the requesting client, the server $\mathcal{S}$ decrypts $\{A_i\}_{B_i}$ using B_i as the decryption key. The request will only be granted if the decrypted message equals A_i and matches the public key of in the server's authorization list. This relies on the fact that only the genuine client C_i has the secret key B_i. Hence, only client c_i can generate the $[A_i]_{B_i}$ which, when decrypted using B_i, will produce the same A_i as the copy stored by the server. A random server challenge will then be performed. The challenge eliminates the possibility of a fake client using another client's certificate obtained by eavesdropping. The server generates a random message, sends it to proxy, which in turn sends it to client C_i. Client C_i encrypts the message using its own private key and replies to proxy.

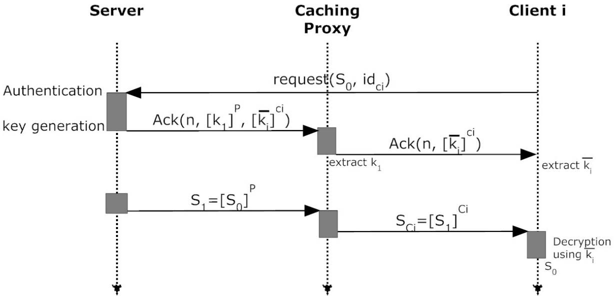

Server
Caching Proxy
Client i
request(S_0, id_{ci})
Authentication
key generation
Ack(n, $[k_1]^P$, $[\overline{k_i}]^{ci}$)
Ack(n, $[\overline{k_i}]^{ci}$)
extract k_1
extract $\overline{k_i}$
$S_1=[S_0]^P$
$S_{Ci}=[S_1]^{Ci}$
Decryption using $\overline{k_i}$
S_0

Fig. 4. Operations for a non cached data

Proxy $\mathcal{P}$ also encrypts the message using its own private key, and includes the encrypted message in its reply. Finally, server $\mathcal{S}$ decrypts the challenge-replies using the public keys of client C_i and proxy $\mathcal{P}$, respectively. A request will only be granted if the decrypted messages match the challenge message. If the authentication is successful, then the server will proceed to the key generation operation.

key Generation. The server randomly generates two large prime numbers p and q, and computes the modulus $n = p.q$, and $\phi = (p-1).(q-1)$, it generate the encryption key k_0, re-encryption key k_i, and the corresponding decryption key $\bar{k}_i$ using RSA algorithm. The server $\mathcal{S}$ then replies back to the proxy $\mathcal{P}$ with the re-encryption key k_0, which is encrypted using the proxy's public key, and the corresponding decryption key $\bar{k}_i$, which is encrypted using the client C_i's public key. Note that the proxy cannot extract the decryption key $\bar{k}_i$, but only the client can perform the decryption to extract $\bar{k}_i$. And since the re-encryption key k_i is encrypted using the proxy's public key, no one can reveal it by eavesdropping on the channel between the server and the proxy. Collusion attacks can thus be avoided. Consider two colluding members knowing the re-encryption keys k_i and k_j (by eavesdropping) and decryption keys $\bar{k}_i$ and $\bar{k}_j$, it can be shown that they can compute the secret parameter efficiently ϕ. Once the secret parameter is revealed, the colluding members can derive the proxy's decryption key or any other client's decryption key. Hence, the re-encryption key k_i must be encrypted such that only the proxy can retrieve it.

The proxy $\mathcal{P}$ replies with an acknowledgment back to the client C_i including the encrypted $\bar{k}_i$. The client decrypts using its own private key to retrieve the decryption key $\bar{k}_i$.

data Encryption. The server $\mathcal{S}$ uses the encryption key k_0 and n to encrypt the multimedia data packets. The server then sends the encrypted data packets to the proxy $\mathcal{P}$ via an ordinary and possibly insecure channel. Upon receiving the encrypted data packets, the proxy caches the data without decryption or modification.

data Re-Encryption and delivery. To send the encrypted data S_1 to a client C_i, the proxy $\mathcal{P}$ uses the re-encryption key k_i and to re-encrypt the already encrypted data packets in the cache. The proxy then sends the re-encrypted data packets to the client C_i via an ordinary and possibly insecure channel. Upon receiving the re-encrypted data packets, the client can use the decryption key $\bar{k}_i$ to decrypt the received data packets.

Scenario 2: Cached Data. We consider the case wherein the data are already cached at the proxy. Figure 5 illustrates the operations between the server $\mathcal{S}$ and the proxy $\mathcal{P}$, and the operations between the proxy $\mathcal{P}$ and the client j for requesting data that are already cached by the proxy. These operations are:

authentication. The client j sends a request to the proxy $\mathcal{P}$ with its certificate. $\mathcal{P}$ forward the request to the server $\mathcal{S}$ with a specified unique identifier ID. The

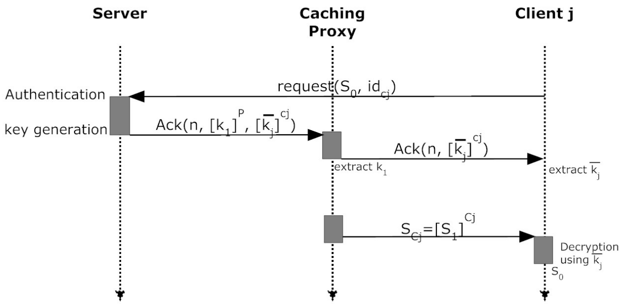

Server
Caching Proxy
Client j
Authentication
key generation
$\text{request}(S_0, id_{cj})$
$\text{Ack}(n, [k_1]^P, [\overline{k_j}]^{cj})$
$\text{Ack}(n, [\overline{k_j}]^{cj})$
extract k_1
extract $\overline{k_j}$
$S_{Cj}=[S_1]^{Cj}$
Decryption using $\overline{k_j}$
S_0

Fig. 5. Operations for a cached data

server needs to authenticate that the request is indeed from the client C_j. If the authentication succeeds, $\mathcal{S}$ will go on to the key generation operation.

key Generation. The server $\mathcal{S}$ randomly generates a re-encryption key k_j and a corresponding decryption key $\bar{k}_j$ based on the ϕ, n, and k_0. It then sends back to $\mathcal{P}$ the re-encryption key k_j, which is encrypted using the proxy's public key, and the decryption key $\bar{k}_j$, which is encrypted using the client C_j's public key. Again, the proxy cannot extract the decryption key $\bar{k}_j$.

$\mathcal{P}$ replies back to the client j with the encrypted decryption key. The client j decrypts using its own private key to retrieve the decryption key $\bar{k}_j$.

data Re-Encryption and Delivery. The proxy $\mathcal{P}$ uses the re-encryption key k_i and n to re-encrypt the cached data packets. It then sends the re-encrypted data packets to the client j via an ordinary and possibly insecure channel. Upon receiving the re-encrypted data packets, the client can use the decryption key $\bar{k}_j$ to decrypt the received data packets.

4 Implementation

The proposed architecture has been implemented using the JAVA technology and currently includes three profiles of content consumers, an encryption proxy, an authentication server and two content servers, each deployed on a separate machine.

Cryptographique support is provided by Java Cryptographic Architecture (JCA) and a third-party BouncyCastele security provider.

We have tested some experiments on an Intel P4 2400Mhz as cache proxy. Network communication between nodes and associated proxy was conducted over a 802.11g wireless network with a speed of 1Mbps. The unicast routing protocol used is Dynamic Source Routing (DSR).

The demo architecture currently features video content delivery services. A VLC server and player has been integrated and made accessible through web

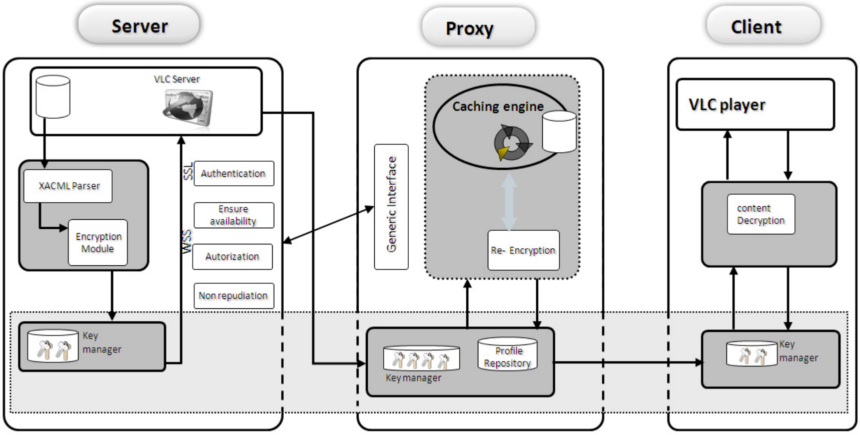


Fig. 6. SPC system architecture

service interfaces. The authentication server features Kerberos ticketing for SSL authentication. All security requirements for the proxy encryption are fulfilled in the proof-of concept architecture. Through a client Gui, including the web service proxies, one can search a videostream and start the downloading if the content is available in the server side or in the cache proxy side.

Figure 6 illustrates the proposal demo architecture for proxy caching over a MANET. Within this architecture we use an SSL/TLS connection with one-way X-509 server authentication, combined with WSS containing a password token inside the header block to be used between the client and the server.

4.1 Performance

The rationale behind the design of our framework was to take into account MANETs constraints to increase reliability and reduce communication cost and delay. The overhead of both channel security and encryption has been analyzed through a number of experiments, Figure 7 shows the delay of sending one video of 3,56Mb passing the cache proxy, without router node, passing 1, 2 and 3 router nodes, with and without content encryption for each case.

The different results show that the encryption impact the delivery delay with about 3 sec. However, using caching proxy enhance this and reduce the delivery delay with more than 3 sec. Moreover this solution reduces the bandwidth cost between proxies and servers using cache techniques. We can say than that this solution reduces the cost of our mobile network.

These preliminary results have to be extended by implementing some optimizations for decryption operations, further verifying access control and authorization, and comparing our performance with other result found in the literature.

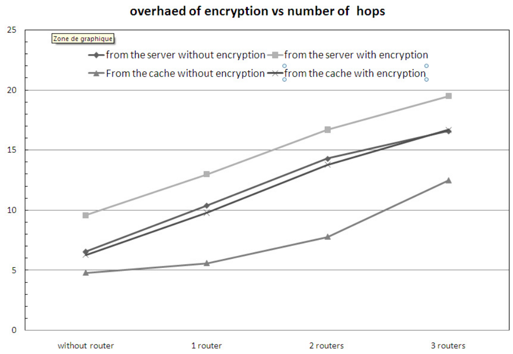


Fig. 7. Average of delay for different type of delivery vs number of hops

5 Conclusion

The area of ad hoc network security has been receiving increasing attention among researchers in recent years. However, little has been done so far in terms of the definition of security needs specific to different types of scenario that can be defined for ad hoc networks. We introduced a scenario of caching proxies and content protection using encryption where there is no shared *a priori* trust between the mobile nodes.

In this paper we have presented an encryption scheme based in a proxied MANET approach, taking into account the wireless and mobility constraints. We exploited in our solution proxy encryption to allow intermediate routers to transform the ciphertext without revealing the secret key and the plaintext. By giving proper conversion keys to intermediate routers, the impacts of changing membership events are confined in a local area. Thus we achieved the goal of containment and scalability. Therefore, our scheme is scalable for large and dynamic mobile end-users.

We noticed some drawbacks in our solution specially in the decryption step. Our next challenge is to optimize these part to have a better results.

References

1. Bresson, E., Chevassut, O., Pointcheval, D.: A Security Solution for IEEE 802.11s Ad-hoc Mode: Password-Authentication and Group-Diffie-Hellman Key Exchange. International Journal of Wireless and Mobile Computing (2005)
2. Burnside, M., Clarke, D., Mills, T., Maywah, A., Devadas, S., Rivest, R.: Proxy-based security protocols in networked mobile devices. In: Proceedings of the 2002 ACM symposium on Applied computing, pp. 265–272 (2002)
3. Buttyán, L., Vajda, I.: Towards provable security for ad hoc routing protocols. In: Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks, pp. 94–105 (2004)
4. Endler, M., Rubinsztejn, H., da Rocha, R., do Sacramento Rodrigues, V.: Proxy-based Adaptation for Mobile Computing. In: PUC (2005)

5. Huang, C., Lee, C.: Layer 7 Multimedia Proxy Handoff Using Anycast/Multicast in Mobile Networks. IEEE Transactions on Mobile Computing 6(4) (2007)
6. Kachirski, O., Guha, R.: Effective intrusion detection using multiple sensors in wireless ad hoc networks. In: Proceedings of the 36th Annual Hawaii International Conference on System Sciences, p. 8 (2003)
7. Lampson, B., Rivest, R.: SDSI— a simple distributed security infrastructure. In: DIMACS Workshop on Trust Management in Networks, South Plainfield, NJ (September 1996)
8. Liu, J., Xu, J.: Proxy caching for media streaming over the Internet. Communications Magazine 42(8), 88–94 (2004)
9. Molva, R., Pannetrat, A.: Scalable Multicast Security with Dynamic Recipient Groups. ACM Transactions on Information and System Security 3(3), 136–160 (2000)
10. Pearce, C., Ma, V., Bertok, P.: A Secure Communication Protocol for Ad-Hoc Wireless Sensor Networks. In: IEEE International Conference on Intelligent Sensors, Sensor Networks & Information Processions, Melbourne, Australia (2004)
11. Rejaie, R., Yu, H., Handley, M., Estrin, D.: Multimedia proxy caching mechanism for quality adaptive streamingapplications in the Internet. In: Proceedings Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies, IEEE, p. 2 (2000)
12. Vaidya, N.: Mobile Ad Hoc Networks: Routing, MAC, and Transport Issues. In: ACM MOBICOM 2001, tutorial (2001)
13. Van Der Merwe, J., Dawoud, D., McDonald, S.: A survey on peer-to-peer key management for mobile ad hoc networks. ACM Computing Surveys 39(1) (2007)
14. Yeun, S., Lui, J., Yau, D.: A multi key secure multimedia using asymmetric reversible parametric sequences: theory, design, and implementation. IEEE Transactions on Multimedia 7(2), 330–338 (2005)

Secure Group Communication with Self-healing and Rekeying in Wireless Sensor Networks

Firdous Kausar[1], Sajid Hussain[2], Jong Hyuk Park[3], and Ashraf Masood[1]

[1] College of Signals, National University of Science and Technology (NUST),
Rawalpindi, Pakistan
firdous.imam@gmail.com, ashrafm61@gmail.com

[2] Jodrey School of Computer Science, Acadia University, Nova Scotia, Canada
sajid.hussain@acadiau.ca

[3] Department of Computer Engineering, Kyungnam University, Masan, Korea
parkjonghyuk@gmail.com

Abstract. We have developed a self-healing key distribution scheme for secure multicast group communications for wireless sensor network environment. We present a strategy for securely distributing rekeying messages and specify techniques for joining and leaving a group. Access control in multicast system is usually achieved by encrypting the content using an encryption key, known as the group key (session key) that is only known by the group controller and all legitimate group members. In our scheme, all rekeying messages, except for unicast of an individual key, are transmitted without any encryption using one-way hash function and XOR operation. In our proposed scheme, nodes are capable of recovering lost session keys on their own, without requesting additional transmission from the group controller. The proposed scheme provides both backward and forward secrecy. We analyze the proposed scheme to verify that it satisfies the security and performance requirements for secure group communication.

Keywords: sensor networks, security, key distribution, secure group communication, one-way hash chains.

1 Introduction

Wireless sensor network (WSN) consists of a large number of small, low cost sensor nodes which have limited computing and energy resources. Usually, sensor nodes perform in-network processing by reducing large streams of raw data into useful aggregated information. Therefore, compromised nodes could deviate the network behavior by injecting false data or modifying data of correct nodes. Thus, it must be guaranteed that compromised nodes do not take part in the group communication.

Secure group communication needs a secret shared by all the group members for group oriented applications in wireless sensor networks (WSNs). The shared key provides group secrecy and source authentication. A single symmetric key known only to the group members can effectively protect a multicast group.

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 737–748, 2007.

However, only legitimate users should have access to the group communication in order to achieve privacy [1]. In rekeying, the session keys are updated periodically, when new users join or old users leave the group. The keys are securely redistributed to the existing members of the group in order to provide forward secrecy (FS) as well as backward secrecy (BS). The newly joined users should not be able to derive the previous group keys, even if they are able to derive future group keys with subsequently distributed keying information. Similarly, the revoked users should not be able to derive the future session keys, even if they are able to compute the previous session keys with previously distributed keying information.

The rekeying is performed by the group controller (GC). The most important parameters when performing group rekeying are as follows: the number of keys stored by the group controller, the number of keys stored by each group member, the number of keys delivered in the initialization stage, bandwidth required for updating the keys, and latency for updating the session key [2].

As the size of the group grows and/or the rate of membership change increases, the frequency of rekeying becomes the primary bottleneck for rekeying on each membership change. Therefore, scalable group rekeying is an important and challenging problem to be addressed in order to support secure multicast communication.

Another important problem in multicast communication is reliability. Since multicasting is an unreliable mode of communication, packets may be lost during the communication. If a packet containing key updating information is lost, authorized receivers may not be able to calculate the session key. This may influence rekeying and so the rekeying system must be self-healing if packet loss occurs. In a large and dynamic group communication over an unreliable network, the main concept of self-healing in key distribution schemes is that users can recover lost session keys on their own, without requesting additional transmissions from the group manager, even if some previous key distribution messages are lost. This reduces network traffic, the risk of user exposure through traffic analysis, and the work load on the group manager.

The key idea of self-healing key distribution schemes is to broadcast information that is useful only for trusted members. Combined with its pre-distributed secrets, this broadcast information enables a trusted member to reconstruct a shared key. On the contrary, a revoked member is unable to infer useful information from the broadcast. The only requirement that a user must satisfy to recover the lost keys through self-healing is its membership in the group both before and after the sessions in which the broadcast packet containing the key is sent. A user who has been off-line for some period is able to recover the lost session keys immediately after coming back on-line. Thus self-healing approach of key distribution is stateless.

The need to form a group might be driven by the query being propagated through a node. As a result, a node may need to define a multicast group to make the query initiated in those nodes and then collect the result efficiently and securely. Further, a node may also modify such queries effectively over the time. For instance, a multicast group could be a region defined with a geometric shape.

This paper provides a computationally secure and efficient group key distribution scheme with self-healing property and time-limited node revocation capability for large and dynamic groups over insecure WSNs. The session keys are updated periodically, where the update is performed regardless of changes in network (group) topology. Periodic rekeying can significantly reduce both the computation and communication overhead at the GC and the nodes, and thus improve the scalability and performance of key distribution protocols. It is shown that the proposed scheme can tolerate high channel loss rate, and hence make a good balance between performance and security, which is suitable for WSN applications.

The paper is organized as follows. In Section 2, related research is described. In Section 3, we describe the preliminaries assumed throughout the paper. We describe the security properties in Section 4, and give details of our proposed scheme in Section 5. In Section 6, we present an analysis of proposed scheme. Finally, we summarize our paper in Section 7.

2 Related Work

Recently there have been several proposals to address the secure group communication issues. The most known technique is the construction of a logical key tree where group members are associated with leaves and each member is given all the keys from his leaves to the root, as proposed in [3] [4] [5] [6], where root key is the group key. This approach allows reducing the communication cost for key update, on the event of group membership change, to $O(logM)$ where M is the number of group members.

Several extensions are proposed to deal with reliability [7], node dependent group dynamics [8], and time variant group dynamics [9]. Extensions to wireless networks are discussed in [10] and several secure multicast protocols are proposed in [11] [12].

Park et al. [13] propose a lightweight security protocol(LiSP) for efficient rekeying in dynamic groups. LiSP utilizes broadcast transmission to distribute the group keys and uses one-way key chains to recover from lost keys. While this scheme is very efficient, LiSP requires the use of static administration keys to perform periodic administrative functions. This leaves those keys vulnerable to disclosure.

Wong et al. [14] propose the the group re-keying, which relies only on current rekeying message and the node's initial configuration. A non-revoked node can decrypt the new session keys independently from the previous re-keying messages without contacting the GC, even if the node is off-line for a while. They use keys of multiple granularity to reduce the rekeying overhead associated with membership management.

Carman et al. [15] give a comprehensive analysis of various group key schemes and find that the group size is the primary factor that should be considered when choosing a scheme for generating and distributing group keys in a WSN.

Staddon et al. [16] propose a self-healing group key distribution scheme based on two-dimension t-degree polynomials. Liu et al. [17] further improve the work in [16] by reducing the broadcast message size in situations where there are frequent but short-term disruptions of communication, as well as long-term but infrequent disruptions of communication. Blundo et al. [18] also present a design of self-healing key distribution schemes which enables a user to recover from a single broadcast message where all keys are associated with sessions where it is a member of the communication group.

Jiang et al. [19] propose a key distribution scheme with time-limited node revocation based on dual directional hash chains for WSNs. Dutta et al. [20] propose two constructions for self-healing key distribution based on one-way hash key chains with t revocation capability using polynomial based node revocation.

3 Preliminaries

Definition 1. *A one-way hash function H can map an input M of the arbitrary length to an output of the fixed length, which is called hash value $h : h = H(M)$, where the length of M is m-bits. One-way hash function H has the following properties [21]:*

- *Given a hash value h, it is computationally infeasible to find the input M such that $H(M) = h$*
- *Given an input M, it is computationally infeasible to find a second input $\acute{M}$ such that $H(\acute{M}) = h$, where $\acute{M} \neq M$*

Definition 2. *Let H be a one-way hash function and s be a random seed. Then, a hash chain can be deduced by iteratively hashing s, which can be written as: $H^i(s) = H(H^{(i-1)}(s)), i = 1, 2, \ldots$ where, s is regarded as "trust anchor" of the one-way hash chain. The hash chain includes a sequence of hash values, which can be denoted by $h_1 = H(s), h_2 = H(h_1), \ldots, h_i = H(h_{i-1}), i = (1, 2, \ldots)$*

Definition 3. *Let $G(x, y) = H(x) \oplus y$, where H is a one-way hash function and $\oplus$ denotes the bitwise XOR. Given x and $G(x, y)$, without the knowledge of y, it is computationally infeasible to find $\acute{y}$ such that $G(x, \acute{y}) = G(x, y)$*

Node Revocation. The concept of node revocation can be described as follows. Let G be the set of all possible group nodes, and R be the set of revoked nodes, where $R \subseteq G$. The group node revocation is required to offer a secure way for GC to transmit rekeying messages over a broadcast channel shared by all nodes so that any node $n_i \in \{G - R\}$ can decrypt the rekeying messages, whereas any node in R, $n_i \in R$, cannot decrypt rekeying messages.

Session Key Distribution with Confidentiality. The confidentiality in the session key distribution requires that for any node n_i, the session key is efficiently determined from the personal secret of n_i and the broadcasted rekeying message from GC. However, for any node in R it is computationally infeasible to determine the session key. What any node n_i learns from broadcast rekeying message, it cannot be determined from broadcasts or personal keys alone. Let a group of k nodes is defined as $n_1, n_2, \ldots, n_k$. If we consider separately either the set of m broadcasts $\{B_1, \ldots, B_m\}$ or the set of k personal keys $\{S_1, \ldots, S_k\}$, it is computationally infeasible to compute session key SK_j (or other useful information) from either set.

Let m denote the total number of sessions in the life cycle of the group communication. Each node is assigned a pre-arranged life cycle (t_1, t_2), which depends on the time of joining. In other words, it can be said that each node will participate in the group communication for $k = t_2 - t_1$ number of sessions. Due to which, once a node's life cycle is expired, it is automatically detached from the group session without requiring the direct intervention of the GC.

For a group with life cycle $(0, m)$, the group key for session j is as follows:

$$SK_j = K_j^F + K_{m-j+1}^B \tag{1}$$

where K_j^F is the forward key and K_{m-j+1}^B is the backward key for session j.

4 Security Properties

A rekeying scheme should provide the following types of security.

Definition 4. *A rekeying protocol provides forward secrecy if for any set $R \subseteq G$, where all $n_l \in R$ are revoked before session j, it is computationally infeasible for the members in R to get any information about SK_i for all $i \geq j$, even with the knowledge of session keys $\{SK_1, \ldots, SK_{j-1}\}$ before session j.*

Definition 5. *A rekeying protocol provides backward secrecy if for any set $J \subseteq G$, where all $n_l \in J$ are newly joined nodes after session j, it is computationally infeasible for the members in J to get any information about SK_i for all $i \leq j$, even with the knowledge of group keys $\{SK_{j+1}, \ldots, SK_m\}$ after session j.*

Definition 6. *A rekeying protocol is key-independent, if it is both forward-secret and backward-secret.*

5 Our Proposed Scheme

In this section, we provide the details of our proposed scheme of self-healing key distribution with time limited node revocation capability. First, nodes are divided into groups, where each group is managed by the group controller (GC). However, the groups are dynamic and regrouping is done after specific duration. The details of group formation is not discussed in this paper. The group life cycle is given by m, which determines the total number of sessions for a group. The GC uses the pseudorandom number generator (PRNG) of a large enough period to produce a sequence of m random numbers $(r_1, r_2, \ldots, r_m)$. The GC randomly picks two initial key seeds, the forward key seed S^F and the backward key seed S^B. In the pre-processing time, it computes two hash chains of equal length m by repeatedly applying the same one-way hash function on each seed. For $K_0^F = S^F$ and $K_0^B = S^B$, the hash sequences are generated as follows:

$$\{K_0^F, H(K_0^F), \ldots, H^i(K_0^F), \ldots, H^{m-1}(K_0^F), H^m(K_0^F)\}$$

$$\{K_0^B, H(K_0^B), \ldots, H^i(K_0^B), \ldots, H^{m-1}(K_0^B), H^m(K_0^B)\}$$

During the initial configuration setup, each node n_i is first assigned a prearranged life cyle (t_1, t_2) where $t_1 \geq 1$ and $t_2 \leq m$. n_i will participate in the group communication $k = t_2 - t_1 + 1$ number of sessions. The node n_i joins the group at time t_1 in session p and will have to leave the group at time t_2 in session q, where $q > p$.

The node n_i receives its personal secret from GC consisting of: 1) a forward key in session p i.e. K_p^F, and 2) k number of random numbers corresponding to the sessions in which node n_i will participate in the group communication. Further, GC securely sends the personal secret to n_i using key encryption key KEK_i shared between n_i and GC, as shown below:

$$GC \rightarrow n_i : E_{KEK_i}(K_p^F, (r_p, r_{p+1}, \ldots, r_q)), MAC(K_p^F \| (r_p, r_{p+1}, \ldots, r_q))$$

The node n_i decrypts the message by its corresponding KEK_i to retrieve its secret. In the j-th session the GC locates the backward key K_{m-j+1}^B in the backward key chain and computes the broadcast message

$$B_j = G(K_{m-j}^B, r_j) \tag{2}$$

When the nodes receive the broadcast message B_j, the session key is generated as follows:

- First, when any node n_i in the group receives the broadcast message, it recovers the backward key K_{m-j+1}^B for session j from B_j, by applying XOR on both B_j and r_j, as given below:

$$K_{m-j+1}^B = B_j \oplus r_j \tag{3}$$

From Equation 2 and Equation 3:

$$K_{m-j+1}^B = G(K_{m-j}^B, r_j) \oplus r_j \tag{4}$$

By substituting the value of G i.e. $G(x, y) = H(x) \oplus y$, backward key is given as follows:

$$K_{m-j+1}^B = H(K_{m-j}^B) \oplus r_j \oplus r_j \tag{5}$$

The backward key is obtained:

$$K_{m-j+1}^B = H(K_{m-j}^B) \tag{6}$$

- Second, the node n_i computes the $j-th$ forward key by applying one-way hash function on its forward key K_p^F as follows:

$$K_j^F = H^{j-p}(K_p^F) \tag{7}$$

- Finally, the node n_i computes the current session key SK_j as follows:

$$SK_j = K_j^F + K_{m-j+1}^B \tag{8}$$

5.1 Adding a Group Member

When a node n_i wants to join an active group, the corresponding actions (steps) are described as follows:

- The node n_i obtains the permission to attach to the group communication from the GC. If it is successful, n_i establishes a common secret key KEK_i shared with the GC.
- GC assigns a life cycle to n_i i.e. t_1, t_2.
- Then, the GC sends the current system configuration to n_i using an Init-GroupKey message, as shown below:

$$GC \rightarrow n_i : E_{KEK_i}(K_p^F, (r_p, r_{p+1}, \ldots, r_q)), MAC(K_p^F \| (r_p, r_{p+1}, \ldots, r_q))$$

  where K_p^F and $(r_p, r_{p+1}, \ldots, r_q)$ are shared personal secret for n_i with life cycle (t_1, t_2)
- Upon receiving the broadcast message from GC, n_i computes the current session key and participates in the network communication.

5.2 Node Revocation

A node with a life cycle (t_1, t_2) detaches from the group at time t_2. Figure 1 shows a time line to illustrate node life cycle and revocation. The node participates only between t_1 and t_2, where the interval is divided into a l number of sessions. Further, as each node is assigned with a $l = t_2 - t_1 + 1$ number of random numbers, it cannot derive the session keys $SK_t = K_t^F + K_{m-t+1}^B$ for $t < t_1$ and $t > t_2$. These l random numbers correspond to the sessions in which the node participate in the group. So, these random numbers can be used for specified sessions only and cannot be used for the remaining sessions. In order to recover K_{m-t+1}^B at time t from the B_t, it requires r_t, which is not available. Thus, a time limited node revocation is achieved implicitly without any intervention from the GC. As a result, the communication and the computation overhead on the GC and group nodes are remarkably reduced.

Compromised Node. If a compromised node is detected, all nodes are forced to be re-initialized. Let n_i with life cycle (t_1, t_2) is compromised in session k, where $t_1 < k < t_2$, as shown in Figure 1. The GC re-initializes the group communication system by re-computing a new random number sequence of length $t_2 - k + 1$ and then unicast it to all group nodes securely.

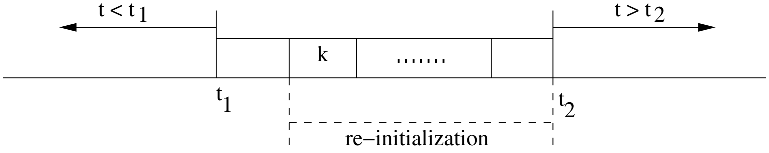


Fig. 1. Node Revocation

6 Analysis

In this section we show that the proposed scheme realizes self-healing key distribution scheme with time limited revocation capability. Further, the forward and backward secrecy is assured with the time-limited node revocation.

We consider a group of sensor nodes as G, $G = \{n_1, n_2, \ldots, n_N\}$, R represents a set of revoked nodes $R \subseteq G$, J represents a set of newly joining nodes $J \subseteq G$, and m is the total number of sessions in the group life cycle.

6.1 Self-healing Property

Consider a node $n_k \in G$ with life cycle (t_1, t_2), which means that n_k joins the group at t_1 (session p) and leaves the group at time t_2 (session q), where $1 \leq p \leq q$, as shown in Figure 2.

Suppose node n_k goes offline in session $p+1$ and comes online again in session $p+j$ where $(p+j) < q$, as shown in Figure 2. As a result, the node n_k will miss the broadcast messages $B_{p+1} \cdots B_{p+j-1}$ from GC; hence, the session keys $SK_{p+1} \cdots SK_{p+j-1}$ will not be available. When node n_k comes online in session $p+j$, it receives the broadcast message B_{p+j} from GC and recovers the backward key $K^B_{m-(p+j)+1}$ for session $p+j$. So, it can obtain the sequence of backward keys $\{K^B_{m-(p+j-1)+1} \ldots K^B_{m-(p+1)+1}\}$ by repeatedly applying H on $K^B_{m-(p+j)+1}$. The node n_k also holds the forward key $K^F_p = H^p(K^F_0)$ of the session p, and hence can obtain the sequence of forward keys $\{K^F_{p+1}, \ldots, K^F_{p+j}\}$ by repeatedly applying H on K^F_p. Now n_k can find all the session keys from session $p+1$ to session $p+j$ without requiring any extra information from GC.

6.2 Key Independence

The proposed scheme also meets the security requirement for forward and backward secrecy, which gives key independence. Informally, forward-secrecy means that the compromise of one or more secret keys does not compromise previous secret keys. Likewise, backward-secrecy refers to that the compromise of one or more secret keys does not compromise future secret keys. Key-independence means that the secret keys used in different sessions are basically independent. Thus, even if the attacker finds out the secret key of a certain session, it does not give any advantage in finding the secret keys of other sessions.

Session #	Forward key	Backward Key	
1	$H(K_0^F)$	$H^m(K_0^B)$	
2	$H^2(K_0^F)$	$H^{m-1}(K_0^B)$	
$\vdots$	$\vdots$	$\vdots$	
p	$H^p(K_0^F)$	$H^{m-(p)+1}(K_0^B)$	$\longleftarrow t_1$
$p+1$	$H^{p+1}(K_0^F)$	$H^{m-(p+1)+1}(K_0^B)$	offline
$\vdots$	$\vdots$	$\vdots$	
$p+j-1$	$H^{p+j-1}(K_0^F)$	$H^{m-(p+j-1)+1}(K_0^B)$	
$p+j$	$H^{p+j}(K_0^F)$	$H^{m-(p+j)+1}(K_0^B)$	online
$\vdots$	$\vdots$	$\vdots$	
q	$H^q(K_0^F)$	$H^{m-(q)+1}(K_0^B)$	$\longleftarrow t_2$
$\vdots$	$\vdots$	$\vdots$	
$m-1$	$H^{m-1}(K_0^F)$	$H^2(K_0^B)$	
m	$H^m(K_0^F)$	$H(K_0^B)$	

Fig. 2. Self-healing in node life cycle

Forward Secrecy. It is shown that a single revoked node or a collusion of revoked nodes cannot learn anything about the future group keys since the secrets they knew while they were authorized member of the group will no longer be used in any future rekeying message.

Let R be the set of revoked nodes and all nodes $n_k \in R$ are revoked before the current session j. The node n_k cannot get any information about the current session key SK_j even with the knowledge of $\{SK_i, SK_{i+1}, \ldots, SK_{j-1}\}$ before session j, where i is the earliest session of all the nodes in R, or in other words, i is the minimum for all t_1's of nodes in R. In order to find SK_j, node n_k needs the random number r_j of that session and that r_j will not be available to n_k. Also, because of the one-way property of H, it is computationally infeasible to compute $K_{j_1}^B$ from $K_{j_2}^B$ for $j_1 < j_2$. The nodes in R may know the sequence of backward keys $K_m^B, \ldots, K_{m-j+2}^B$; however, they cannot compute K_{m-j+1}^B in order to find current session key SK_j.

Backward Secrecy. Let J is the set of nodes that join the group in session j. The collusion of newly joining nodes cannot get any information about any previous session keys before session j even with the knowledge of group keys after session j. Each $n_k \in J$ when joins the group, GC gives it $j-th$ forward key i.e. K_j^F, instead of initial forward seed K_0^F. As $K_j^F = H(K_{j-1}^F)$, it is computationally infeasible for n_k to compute the previous forward keys which are required to compute session keys before current session j. Hence, the proposed scheme is backward secure.

Table 1. Time and memory requirements for Tmote Sky

Algorithm	Time (seconds)	RAM (bytes)	ROM (bytes)	Energy (Joules)
SHA-1	10.545×10^{-3}	128	$4,048$	56.94×10^{-6}
MD5	5.757×10^{-3}	176	$12,500$	31.09×10^{-6}

6.3 Storage Requirements

In our scheme the GC and all nodes do not need any encryption/decryption process of a re-keying message to update session keys. All computation needed for re-keying is one-way hash function and XOR operation, and all information needed for re-keying is in the current transmission and the initial information.

We implement two one-way hash algorithms, SHA-1 and MD5 using nesC [22] programming language in TinyOS for Moteiv's Tmote Sky sensors. We have considered voltage level of 3 volts and nominal current (with Radio off) as 1.8×10^{-3} amps, as given in Tmote Sky's data sheet [23]. We take data stream of 64 bytes. As shown in Table 1, for SHA-1 the code consumes 128 bytes of RAM, 4048 bytes of ROM, takes approximately 10.5 ms to produce a 160-bit hash of a 64-byte message, and the energy consumption is 56.94 μJoules. MD5 produces a 128-bit message digest for a given data stream. The code consumes 176 bytes of RAM, 12.5 KB of ROM, takes approximately 5.75 ms to hash a message of 64 bytes using 64-byte blocks, and the energy consumption is 31.09 μJoules. The above implementation shows that SHA-1 consumes less memory than MD5; however, it's processing overhead is almost double than MD5.

7 Conclusion

Efficient solutions for the problem of key distribution are essential for the feasibility of secure group communication in sensor networks. In this paper, we develop a key distribution scheme for secure group communication in WSNs. The scheme provides a self-healing mechanism for session key-recovery on possible packet loss in the lossy environment using one-way key chain.

Other features include periodic re-keying of group key and time-limited group node revocation. The session keys are updated periodically, where the update is performed regardless of changes in network (group) topology. Periodic rekeying significantly reduces both the computation and communication overhead at the GC and the nodes, and thus improves the scalability and performance of the proposed scheme. Further, the time-limited node revocation is achieved without any intervention from the GC.

The analysis shows that the proposed scheme is computationally secure and meets the security requirements for forward and backward secrecy. The implementation of two one-way hash algorithms SHA-1 and MD5 on resource constraint sensor nodes (Tmote Sky) shows the feasibility of the proposed scheme for current wireless sensor network technology. Hence, the scheme results scalable, and particularly attractive for large dynamic groups.

Acknowledgment

This work is in part supported by Higher Education Commission (HEC) Pakistans International Research Support Initiative Programs scholarship given to Firdous Kausar to conduct her research at Acadia University, Canada. Further, we would like to thank National Science and Engineering Research Council (NSERC) Canada for their support in providing RTI and Discovery grants to Dr. Hussain at Acadia University, Canada. This research is also supported by Kyungnam University.

References

1. Wallner, D., Harder, E., Agee, R.: Key management for multicast: Issues and architectures (1999)
2. Canetti, R., Garay, J., Itkis, G., Micciancio, D., Naor, M., Pinkas, B.: Multicast Security: A Taxonomy and Some Efficient Constructions. In: INFOCOMM 1999 (1999)
3. Kurnio, H., Safavi-Naini, R., Wang, H.: A secure re-keying scheme with key recovery property. In: Proceedings of the 7th Australian Conference on Information Security and Privacy, pp. 40–55. Springer, London, UK (2002)
4. Wang, L., Wu, C.K.: Authenticated group key agreement for multicast. In: The 5th International Conference on Cryptology and Network Security, Springer, Heidelberg (2006)
5. Ki, J.H., Kim, H.J., Lee, D.H., Park, C.S.: Efficient multicast key management for stateless receivers. In: Lee, P.J., Lim, C.H. (eds.) ICISC 2002. LNCS, vol. 2587, pp. 497–509. Springer, Heidelberg (2003)
6. Pegueroles, J., Bin, W., Soriano, M., Rico-Novella1, F.: Group rekeying algorithm using pseudo-random functions and modular reduction. In: Li, M., Sun, X.-H., Deng, Q.-n., Ni, J. (eds.) GCC 2003. LNCS, vol. 3032, pp. 875–882. Springer, Heidelberg (2004)
7. Yang, Y.R., Li, X.S., Zhang, X.B., Lam, S.S.: Reliable group rekeying: a performance analysis. In: SIGCOMM 2001. Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications, pp. 27–38. ACM Press, New York, NY, USA (2001)
8. Poovendran, R., Baras, J.S.: An information theoretic analysis of rooted-tree based secure multicast key distribution schemes. In: Wiener, M.J. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 624–638. Springer, Heidelberg (1999)
9. Noubir, G., Zhu, F., Chan, A.H.: Key management for simultaneous join/leave in secure multicast. In: Proceedings of MILCOM (2003)
10. Gong, L., Shacham, N.: Multicast security and its extension to a mobile environment. Wirel. Netw. 1(3), 281–295 (1995)
11. Bruschi, D., Rosti, E.: Secure multicast in wireless networks of mobile hosts: protocols and issues. Mob. Netw. Appl. 7(6), 503–511 (2002)

12. Kostas, T., Kiwior, D., Rajappan, G., Dalal, M.: Key management for secure multicast group communication in mobile networks. In: Proceedings of DARPA Information Survivability Conference and Exposition (2003)
13. Park, T., Shin, K.G.: Lisp: A lightweight security protocol for wireless sensor networks. Trans. on Embedded Computing Sys. 3(3), 634–660 (2004)
14. Wong, C.K., Gouda, M., Lam, S.S.: Secure group communications using key graphs. IEEE/ACM Trans. Netw. 8(1), 16–30 (2000)
15. Carman, D., Matt, B., Cirincione, G.: Energy-efficient and low-latency key management for msn networks. In: Proceedings of 23rd Army Science Conference, Orlando FL (2002)
16. Staddon, J., Miner, S., Franklin, M., Balfanz, D., Malkin, M., Dean, D.: Self-healing key distribution with revocation. In: Proceedings of IEEE Symposium on Security and Privacy, pp. 241–257 (2002)
17. Liu, D., Ning, P., Sun, K.: Efficient self-healing group key distribution with revocation capability. In: CCS 2003. Proceedings of the 10th ACM conference on Computer and communications security, pp. 231–240. ACM Press, New York, NY, USA (2003)
18. Blundo, C., Darco, P., Santis, A.D., Listo, M.: Design of self-healing key distribution schemes. Des. Codes Cryptography 32(1-3), 15–44 (2004)
19. Jiang, Y., Lin, C., Shi, M., Shen, X.: Self-healing group key distribution with time-limited node revocation for wireless sensor networks. Ad Hoc Networks 5(1), 14–23 (2007)
20. Dutta, R., Chang, E.C., Mukhopadhyay, S.: Efficient self-healing key distribution with revocation for wireless sensor networks using one way key chains. In: ACNS 2007. Proceedings of 5 th International Conference on Applied Cryptography and Network Security (2007)
21. NIST: Secure hash standard. In: National Institute for Standards and Technology, Gaithersburg, MD, USA (April 1995)
22. Gay, D., Levis, P., von Behren, R., Welsh, M., Brewer, E., Culler, D.: The nesc language: A holistic approach to networked embedded systems. SIGPLAN Not. 38(5), 1–11 (2003)
23. http://www.moteiv.com/products/docs/tmote-skydatasheet.pdf

Multigrid Based Key Predistribution Scheme in Ad Hoc Networks

Liu Cong[1] and Jiang Huangpu[2]

[1] College of Science, YanShan University, 066004 QinHuangDao, China
[2] College of Science Information and Engineering, YanShan University ,
066004 QinHuangDao, China
sirengirl@163.com, huangpujiang66@163.com

Abstract. Using polynomial-based key predistribution in distribution sensor networks for reference, this paper presents an efficient key predistribution scheme for Ad hoc networks: a multigrid-based key predistribution scheme. In this scheme we put forward the concept of multi-grid on which our key predistribution scheme is upbuilt, and the first time threshold scheme is introduced into the process of key transfer. The analysis in this paper indicates that this scheme has a number of nice properties, including high probability to establish pairwise keys, tolerance of node captures, and low communication overhea.

Keywords: Bivariate t-degree polynomial, Key management system, Key predistribution scheme, Ad hoc networks.

1 Introduction

A mobile ad hoc network (MANET) is a collection of independent mobile nodes. It offers convenient infrastructure-free communications over the shared wireless channel. In recent years, ad hoc wireless networking has found applications in military, commercial and educational environments such as emergency rescue missions, home networks of personal devices, and instantaneous classroom/ conference room applications. However, unlike their wired counterpart, an infrastructureless ad hoc network is vulnerable to eavesdropping and the nodes in this network often have little physical protection. Examples of such attacks include passive eavesdropping over the wireless channel, denial of service attacks by malicious nodes as well as attacks from compromised nodes or stolen devices. In order to counteract some of these threats, a MANET have to uses key to establish Security Associations (SAs). Thus, the KMS (Key Management System) is at the heart of the networks defenses.

However, portable devices usually have constraints in computational capability, bandwidth and battery power to afford computationally expensive operations. Using certificate-based KMS may not be possible for the network formed by resource constrained portable devices [1]. And some distributed authentication services in ad hoc networks are not practical.

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 749–759, 2007.

In this paper, we develop a key predistribution technique to deal with the above problems. In order to facilitate the study of new key distribution techniques, we first study the general framework for pairwise key establishment based on the polynomial based key predistribution protocol in [2] and the probabilistic key distribution in [3, 4]. All the previous schemes are special instances in this framework. By instantiating the components in this framework, we further develop a novel pairwise key predistribution schemes for ad hoc: multigrid based key predistribution scheme. This scheme has a lot of attractive properties. First, it guarantees that any two sensors can establish a pairwise key when there are no compromised sensors. Second, this scheme is resilient to node compromise. Even if some nodes are compromised, there is still a high probability to establish a pairwise key between non-compromised sensors. Third, a node can directly determine whether it can establish a pairwise key with another node and how to compute the pairwise key. In this paper, word "node" represents Ad Hoc node, because sensor node is represented as "sensor node".

2 Related Works

2.1 Summary

There are several key management schemes for Internet, such as Kerberos、X.509 and PKIX [5]. While this model works well in wired networks, it fails in large-scale MANET [6] for several reasons: (a) Ad hoc networks provide no infrastructure support. The cost of maintaining such centralized servers may be prohibitively high. (b) The CA servers are exposed to single points of compromises and failures. They expose to various malicious attacks. (c) Multihop communications over the error-prone wireless channel expose data transmissions to high loss rate and larger average latency. Frequent route changes induced by mobility also make locating and contacting CA servers in a timely fashion non-trivial.

In [7], the authors propose an (n,t+1) threshold cryptography scheme based distributed public-key management service for ad hoc networks. The private key k is divided into n shares, and the final signature will be computed by the combiner. A more recent proposal[5] describes a similar approach, but it provides a more fair distribution of the burden by allowing any node to carry a share of the private key of the service. The advantage is increased availability, since now any t+l nodes in the local neighborhood of the requesting node can issue or renew a certificate. However, there are some problems with this proposal: First, the number t must be a trade-off between availability and robustness; Second, the system seems to be vulnerable to the Sybil attack. Third, the service needs a initial organization to authorize some CA nodes.

Fallowing the extensive application of DSN (Distribution Sensor Networks) those years, a lot of key management schemes in DSN were put forward. Eschenauer and Gligor proposed a probabilistic key predistribution scheme recently for pairwise key establishment [3]. The main idea was to let each sensor node randomly pick a set of keys from a key pool before deployment so any two sensor nodes have a certain probability of sharing at least one common key. Chan et al. further extended this idea and developed two key predistribution techniques: q-composite key predistribution and random pairwise keys scheme [4]. The q-composite key predistribution also uses a key pool but requires two sensor nodes compute a pairwise key from at least q predistributed keys they share. The random pairwise keys scheme randomly picks pairs of sensor nodes and assigns each pair a unique random key. Both schemes improve the security over the basic probabilistic key predistribution scheme. Donggan Liu and Peng Ning proposed two efficient schemes: a random subset assignment key predistribution scheme and a grid-based key predistribution scheme [8]. Because our scheme is based on grid-based key predistribution scheme, we describe grid-based scheme in detail.

2.2 Grid-Based Key Predistribution

To predistribute pairwise keys, the (key) setup server randomly generates a bivariate t-degree polynomial over a finite field F_q, witch is shown in equation (1).

$$\mathrm{f}\,(\mathrm{x},\mathrm{y}) = \sum_{i,j=0}^{t} a_{ij} x^i y^j \tag{1}$$

In equation (2) q is a prime number that is large enough to accommodate a cryptographic key, such that it has the property of f (x, y) = f (y, x). (In the following, we assume all the bivariate polynomials have this property without explicit statement.) It is assumed that each sensor has a unique ID. For each sensor i, the setup server computes a polynomial share of f (x, y), that is, f (i, y). For any two sensor nodes i and j, node i can compute the common key f (i, j) by evaluating f(i, y) at point j, and node j can compute the same key f(j, i) = f(i, j) by evaluating f(j, y) at point i. In this approach, each sensor node i need to store a t-degree polynomial f (i, x), which occupies (t+1) log q storage space. To establish a pairwise key, both sensor nodes need to evaluate the polynomial at the ID of the other sensor node. There is less communication than other schemes during the pairwise key establishment process.

As shown in Figure 1, they define grid as follow: Supposed a sensor network has at most N sensor nodes. The grid-based key predistribution scheme then constructs an m×m grid with a set of 2m polynomials as it is shown in equation (2), where $\mathrm{m}>\lceil \sqrt{\mathrm{N}} \rceil$.

$$F=\left\{\mathrm{f}_{\mathrm{i}}^{\mathrm{r}}(\mathrm{x},\mathrm{y}),\mathrm{f}_{\mathrm{i}}^{\mathrm{c}}(\mathrm{x},\mathrm{y})\right\}_{\mathrm{i}=0,\ldots,\mathrm{m}-1}. \tag{2}$$

Subset assignment: As shown in Figure 1, each row i in the grid is associated with a polynomial $\mathrm{f}_{\mathrm{i}}^{\mathrm{r}}(\mathrm{x},\mathrm{y})$, and each column i is associated with a polynomial $\mathrm{f}_{\mathrm{i}}^{\mathrm{c}}(\mathrm{x},\mathrm{y})$. The setup server assigns each sensor in the network to a unique intersection in this grid. For the sensor at the coordinate (i, j), the ID of this sensor node is ID $=\langle \mathrm{i},\mathrm{j}\rangle$. The setup server then distributes $\left\{\mathrm{ID},\mathrm{f}_{\mathrm{i}}^{\mathrm{r}}(\mathrm{j},\mathrm{x}),\mathrm{f}_{\mathrm{j}}^{\mathrm{c}}(\mathrm{i},\mathrm{x})\right\}$ to this sensor node. To facilitate path discovery, we require that the intersections allocated to sensor nodes are densely selected within a rectangle area in the grid.

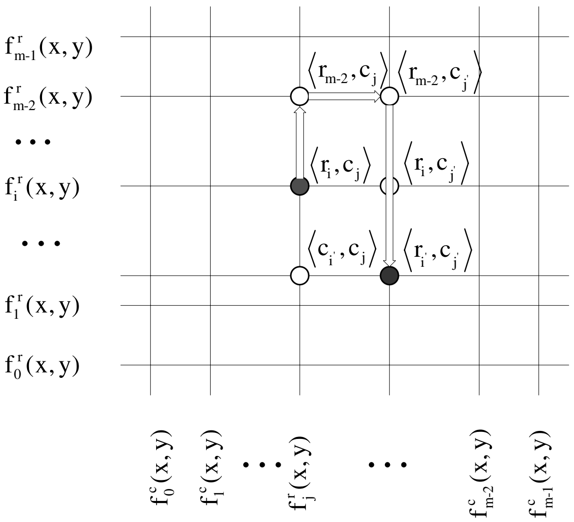

$f^r_{m-1}(x,y)$
$f^r_{m-2}(x,y)$
$f^r_i(x,y)$
$f^r_1(x,y)$
$f^r_0(x,y)$
$\langle r_{m-2}, c_j\rangle$
$\langle r_{m-2}, c_{j'}\rangle$
$\langle r_i, c_j\rangle$
$\langle r_i, c_{j'}\rangle$
$\langle c_{i'}, c_j\rangle$
$\langle r_{i'}, c_{j'}\rangle$
$f^c_0(x,y)$
$f^c_1(x,y)$
$f^r_j(x,y)$
$f^c_{m-2}(x,y)$
$f^c_{m-1}(x,y)$

Fig. 1. Grid

Polynomial share discovery: To establish a pairwise key with sensor node j, sensor node i checks whether $c_i = c_j$ or $r_i = r_j$. If $c_i = c_j$, both sensor nodes i and j have polynomial shares of $f^r_{r_i}(x,y)$, and they can use the polynomial-based key predistribution scheme to establish a pairwise key directly. Similarly, if $r_i = r_j$, they both have polynomial shares of $f^c_{c_i}(x,y)$, and can establish a pairwise key accordingly. If neither of these conditions is true, sensor nodes i and j go through path discovery to establish a pairwise key.

Path Discovery: Sensor nodes i and j need to use path discovery if $c_i \neq c_j$ and $r_i \neq r_j$. In Figure 1, we can establish pairwise key through one sensor node (for example $\left\langle r_{i'}, c_j \right\rangle$ or $\left\langle r_i, c_{j'} \right\rangle$) or serval sensor nodes (for example $\left\langle r_{m-2}, c_j \right\rangle$ and $\left\langle r_{m-2}, r_{j'} \right\rangle$) in the form of key transference.

But this scheme for sensor networks makes use of the characteristic of DSN, such as high sensor nodes density, limit resource, and ephemeral lifecycle. So those schemes will bring much trouble if we use it in Ad Hoc networks directly. First, nodes density of Ad Hoc network is low, so the use of this scheme will bring on low connectivity. Second, because there is key transference, so it is not absolute secure in the key transference. We can see that we need increase the connectivity of Ad Hoc, and enhance the security of Ad Hoc, if we want to apply the schemes for DSN to Ad Hoc. To resolve these problems, we introduce threshold theory into key predistributionn scheme.

3 MultiGrid-Based Key Predistribution

This section presents an efficient key predistribution scheme for Ad hoc networks, and we call it multigrid-based key predistribution.

3.1 Summary

As shown in figure 2, we deifine multigrid as follow: Supposed that an Ad hoc network has at most N nodes, the set of users is U=$\{user, user_2..., user_N\}$, The setup server then constructs a set of grids G=$\{G_1, G_2..., G_W\}$ (the value of W will be discuss in future). Each grid G_k $(G_k \in G)$ is constructed of a set of 2m polynomials that is defined in equation that is defined in equation (2). For$i(i \in U)$, the setup server work out O=$\{e_1,e_2...,e_w \mid e_1 \in G_1, \; e_2 \in G_2..., e_W \in G_W\}$, then it work out the set P of polynomials to i, which decided by the set O. Now, we can discribe the ID of some node at intersection $((i_1, j_1),(i_2, j_2)...,(i_W, j_W))$ as $\langle\langle i_1, j_1\rangle,\langle i_2, j_2\rangle...,\langle i_W, j_W\rangle\rangle$. Or we can describe the ID of it as $\left\langle\left\langle r_{i_1}, c_{j_1}\right\rangle,\left\langle r_{i_2}, c_{j_2}\right\rangle...,\left\langle r_{i_W}, c_{j_W}\right\rangle\right\rangle$, where $\left\langle r_{i_k}, c_{j_k}\right\rangle$ is the segment of ID in G_k $(G_k \in G)$, called ID_k , and r_{i_k} , c_{j_k} are the first and last *l* bits of ID_k , respectively.

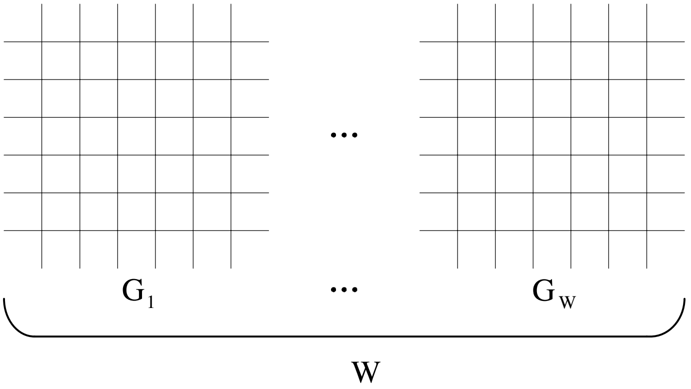


Fig. 2. Multigrid

3.2 Subset Assignment

The setup server first constructs a set of grids G=$\{G_1, G_2..., G_W\}$. Each grid G_k $(G_k \in G)$ is constructed of a set of 2m polynomials (2). For each mobile node $i \in U$, the setup server give a set of intersections O, and get the set P. The setup server then distributes $\{ID, P\}$ to this sensor node. To facilitate path discovery, we require that the intersections allocated to sensors are densely selected within a rectangle area in any grid G_k $(G_k \in G)$.

3.3 Polynomial Share Discovery

To establish a pairwise key with node j, node i need perform polynomial share discovery of grid-based key predistribution scheme on each G_k $(G_k \in G)$: node i checks whether $c_i = c_j$ or $r_i = r_j$. If $c_i = c_j$, both nodes i and j have polynomial shares of $f_{r_i}^{r}(x, y)$, and they can use the polynomial-based key predistribution scheme to establish a pairwise key directly. Similarly, if $r_i = r_j$, they both have polynomial shares of $f_{c_i}^{c}(x, y)$, and can establish a pairwise key accordingly. If there is not $c_i = c_j$ or $r_i = r_j$ in any G_k $(G_k \in G)$, node i and j have to go through path discovery to establish a pairwise key.

3.4 Path Discovery

If two nodes have shares of the same polynomial, they can establish a pairwise key directly. But if two nodes do not have shares of the same polynomial, they must use path discovery to establish a pairwise key. Path discovery of our scheme is completely different from that of the grid-based scheme. As Figure 3 shows, our scheme gets many paths as the routing lookups mechanism does in an on-demand route protocol [9], where real line donates QueryPacket and dashed line donates ReplyPacket. Then it choices k paths that we get to transfer the key shares, which are formed using using the Lagrange polynomial. To transfer key shares securely on different key path, threshold scheme is introduced into our scheme [10, 11]. Finally, the destination node reestablishes the key using the Lagrange polynomial.

Suppose that two nodes have to communicate, but they do not have shares of the same polynomial. The source node A floods a path query packet (PathQuery, shown in the figure 3 with real line). Nodes that receive this PathQuery judge if itself is the destination. If it is, it will send a path reply packet (PathReply, shown in the figure 2 with dashed line) to the source node according to the path information included in the PathQuery. If it is not, it will transmit this PathQuery, and it adds its identity information into the node list of the PathQuery. And ID in PathQuery is used to differentiate this PathQuery with others packets. We must notice that nodes that receive PathQuery can not transmit PathQuery randomly and send PathReply randomly. They must do that according to given way, and these principles will be presented later.

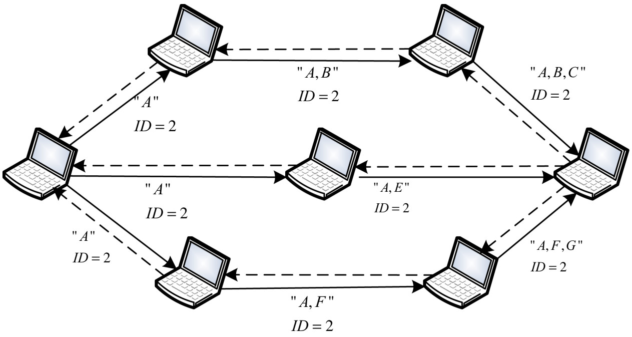
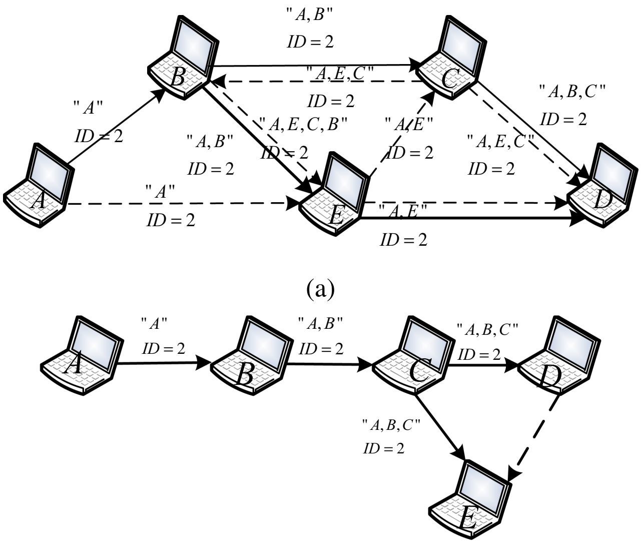


Fig. 3. Path Discovery

To avoid flooding storm and get enough key paths, we must these flooding ways. First, if node that receives PathQuery is not in the node list of the PathQuery, it will flood the PathQuery. Second, if the TTL of PathQuery is not bigger than 0, node that receives PathQuery can not transmit it. As Figure 4(a) shown, node E receives three PathQuery packets. Node E is not in the node lists of packets “A” and “A, B”, so E will transmit them. Node E is in the list of packet “A, E, C, B”, so E will do nothing with it. In Figure 4(b), we proposed that all packets’ TTL are three. Node D do not transmit packet “A, B, C” to E, because packet TTL of “A, B, C” is 0 now.



(a)

(b)

Fig. 4. Flooding

Key node is node that participating in the encryption and decryption process of key share.In figure 5, Node A get $E_{K_C}(\text{ Packet })$ through encrypting “Packet” using key E_{K_C}, And send it to node C. Node C decrypt it and get “Packet”. Then C get $E_{K_E}(\text{ Packet })$ through encrypting “Packet” using key E_{K_E}, And send it to node E. Node E decrypt it and get “Packet”. In this process, node C is key node, but B and D is not.

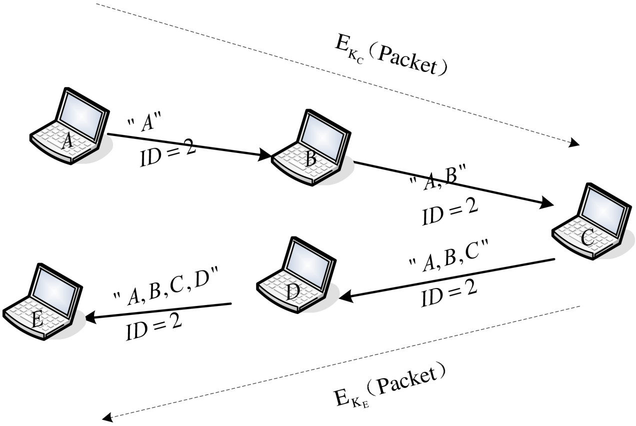


Fig. 5. Key Node

3.5 Key Transference Based on Threshold Theory

The process of key transfer is discussed in detail as follow:

Supposed there is a big prime number p, and a key $k \in K = Z_p$. The source node i randomly establish a t-1 degree polynomial witch is shown in equation (3). In equation (3) a_0=k and $a_1 ..., a_{t-1} \in z_p$ .Then it computes key shares $y_i = h(x_i)$, where x_i $(1 \le i \le n)$ are different nonzero elements from Z_p. We should note that $a_0, a_1 ..., a_{t-1}$ is not public, but $p, x_1, x_2 ..., x_n$ is.

$$h(x) = (a_{t-1}x^{t-1} + ... + a_1 x + a_0) \bmod p \tag{3}$$

The source node transfer y_i $(1 \le i \le n)$ to the destination through k different key paths. If $n \le k$ then each path transfers one key share. If n>k then it’s possible that some nice paths need transfer more than one key share.

If the destination node has received t key shares y_{i_s} $(1 \le s \le t)$, using the Lagrange polynomial, it can reestablish h(x) using the Lagrange polynomial equation (4). Then the destination node can reestablish key using equation (5).

$$h(x)=\sum_{s=1}^{t} y_{i_s}\prod_{j=1,j\neq s}^{t}\frac{x-x_{i_j}}{x_{i_s}-x_{i_j}}. \tag{4}$$

$$k=h(0)=\sum_{s=1}^{t} y_{i_s}\prod_{j=1,j\neq s}^{t}\frac{-x_{i_j}}{x_{i_s}-x_{i_j}}. \tag{5}$$

4 Analysis

4.1 Probability of Share Polynomial

Since each sensor node has 2W polynomial shares and each bivariate polynomial is shared by about 2W (m-1) different sensor nodes, each sensor node can establish a pairwise key with 2W (m– 1) other sensor nodes directly. Thus, among all the other sensor nodes, the percentage p_1 that a node can establish a pairwise key with other nodes directly is $\frac{2W}{m+1}$, witch is defined in equation (6). But p_2 of the grid-based scheme is only $\frac{2}{m+1}$. Figure 6 shows the relationship between the probability p that two nodes share a polynomial and the grid size m. Multigrid-based scheme can establish a pairwise key by higher probability. As the count m of grid larger, the advantage bigger.

$$p_1=\frac{2W(m-1)}{N-1}\approx\frac{2W(m-1)}{m^2-1}=\frac{2W}{m+1}. \tag{6}$$

$$p_2=\frac{2m}{N-1}\approx\frac{2}{m+1}. \tag{7}$$

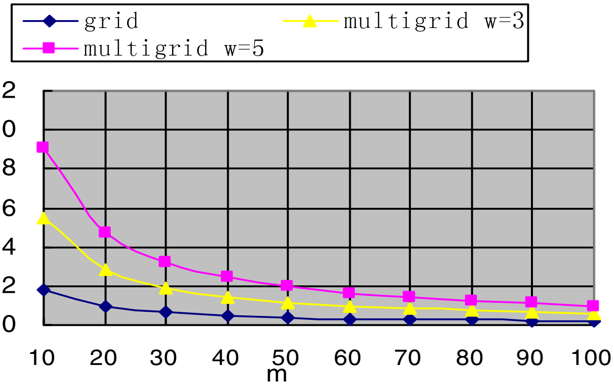


Fig. 6. Probability of two sensors share a polynomial v.s. the grid size m

4.2 Ability to Resist Attack

We assume that the fraction that nodes in the network are compromised is p. Then the probability that exactly i shares on a particular bivariate polynomial have been disclosed is P(i) shown in equation (8).The probability of some bivariate polynomial being compromised is P_C shown in equation (9). For the sake of computation, it supposed that we need an average of k paths to transfer a key, and each path have an average of 3 hops. So the probability that some path is comprised is P_L shown in equation (10), and the probability that some key is comprised is P_K shown in equation (11). Figure 7 shows the relationship between probability to establish a pairwise key and the fraction of compromised nodes. We can see that multigrid based scheme is preponderant distinctly. And as the count of grid larger, the advantage bigger.

$$P(i) = \frac{m\,!}{i!(m-i)!} p^i (1\text{-}p)^{m\text{-}i} . \tag{8}$$

$$P_c = 1 - \sum_{i=0}^{t} P(i) . \tag{9}$$

$$P_L = 1 - (1 - P_C)^3 . \tag{10}$$

$$P_k = (P_L)^k . \tag{11}$$

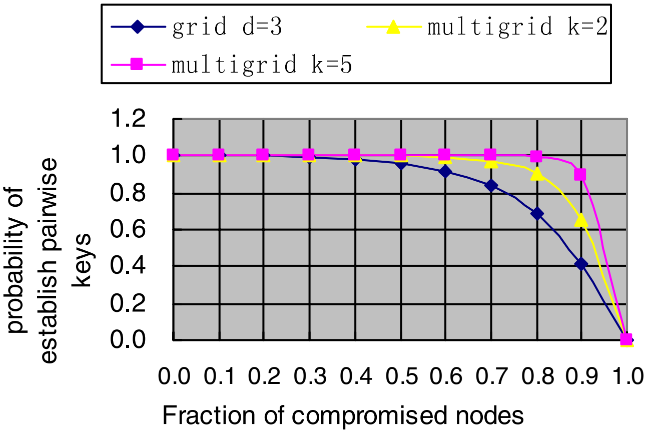


Fig. 7. Probability to establish a pairwise key v.s. the fraction compromised nodes

Acknowledgments. The authors Jiang Huangpu and Wang Xinsheng thank YanShan University for their help. And the author Jiang Huangpu thanks my teacher Sir Wang XinSheng. It is him let me know how to study my project and how to write a paper. The authors would also like to thank all the reviewers for their valuable comments.

References

1. Wang, S.H., Wang, M.Q.: Analytical Overview of Key Establishment and Management for Ad hoc Networks. Application Research of Computer 21, 9–11 (2004)
2. Blundo, C., Santis, A.D., Herzberg, A., Kutten, S., Vaccaro, U.: Perfectly secure key distribution for dynamic conferences. Information and Computation 146, 1–23 (1995)
3. Eschenauer, L., Gligor, V.D.: A key-management scheme for distributed sensor networks. In: Proceeding of the 9th Association for Computing Machinery Conference on Computer and Communications Security, pp. 41–47. ACM Press, Washinigton (2002)
4. Chan, H., Perrig, A., Song, D.: Random key predistribution schemes for sensor networks. In: IEEE Symposium on Research in Security and Privacy, pp. 197–213. IEEE Press, Piscataway (2003)
5. Luo, H., Zerfos, P., Kong, J., Lu, S., Zhang, L.: Self-securing Ad Hoc Wireless Networks. In: IEEE ISCC 2002. Proceedings of the Seventh International Symposium on Computers and Communications, pp. 623–628. IEEE Press, Chicago (2001)
6. Perlman, R.: An overview of PKI trust models. IEEE Networks 13, 38–43 (1999)
7. Zhou, L., Haas, Z.J.: Securing ad hoc networks. IEEE Networks 13, 24–30 (1999)
8. Liu, D., Ning, P.: Establishing pairwise keys in distributed sensor networks. In: Proceedings of 10th ACM Conference on Computer and Communications Security, pp. 52–61. ACM Press, Washington (2003)
9. Perkins, C.E., Bhagwat, P.: Highly dynamic Destination-Sequenced Distance-Vector routing for mobile computers. In: Proceedings of the SIGCOMM 1994 Conference on Communications Architectures, pp. 234–244. IEEE Press, London (1994)
10. Shamir, A.: How to share a secret. Communications of the ACM 22, 612–613 (1979)
11. Chang, T.Y., Yang, C.C., Hwang, M.S.: A Threshold Signautre Scheme for Group Communications without a Shared Distribution Center. Future Generation Computer Systems 20, 1013–1021 (2004)

A Secure Privacy-Preserving Hierarchical Location Service for Mobile Ad Hoc Networks*

Xinghua Ruan, Boyang Yu, Jingdong Xu, and Lin Yang

Department of Computer Science, Nankai University, 300071 Tianjin, China
ruanxinghua@gmail.com, bobyu@mail.nankai.edu.cn,
xujd@nankai.edu.cn, cameling_yang@yahoo.com.cn

Abstract. Recently, position-based routing has proven to be a scalable and efficient way for packet routing in mobile ad hoc networks. To enable position-based routing, a node must be able to discover the location of the intended destination node. This can typically be accomplished by a location service. By far there have been many efficient location service algorithms such as the DREAM, RLS, Homezone, GLS, DLM and HLS, but most of them have focused on the scalability and efficiency of algorithm while the security and privacy issues were vastly neglected. In this article, we propose a secure privacy-preserving hierarchical location service (SPPHLS) based on the HLS algorithm using the broadcast encryption scheme and broadcast authentication scheme. In the proposed secure location service scheme, the position privacy of nodes is protected and the security is promised. Finally, through simulation and analysis, we further show that the proposed scheme only introduces very moderate success rate degradation and query delay compared to the original HLS algorithm.

1 Introduction

Mobile Ad-Hoc Networks (MANETs) are a new wireless networking paradigm for mobile hosts. In a mobile ad-hoc network, there is no fixed infrastructure such as base stations, and a number of autonomous nodes collaborate in order to transport information. Furthermore, the nodes move quickly and the network topology changes frequently. Due to these characteristics, providing scalable and efficient routing schemes for MANETs is quite a challenging task. Recent researches [1, 2] have shown that position-based routing is a well-suited solution for the challenging task in highly dynamic mobile ad hoc networks. Position-based routing protocols can offer significant performance improvements over topology-based routing protocols in large and dense mobile ad hoc networks by using position information to make forwarding decision. According to position-based routing schemes, nodes must be able to discover the intended destination node's position in order to forward the packets. This can be typically accomplished by location service [3]. By far, there have been many efficient

* This work is supported by Tianjin Informatization Foundation under grant No. 061084012 and Microsoft Research Asia Trustworthy Computing Research.

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 760–771, 2007.

location service schemes such as the DREAM, RLS, Homezone, GLS [4], DLM [5] and HLS [6]. But almost all of them focus on the scalability and efficiency of the algorithms while the privacy and security issues were vastly neglected. Malicious nodes can obtain the position information of an arbitrary node through the location service so as to track the node. Thus the position privacy of nodes is badly threatened. Furthermore, the malicious nodes may forge faulty position information to update the location servers so as to influence the communication between other legitimate nodes. However, so far the security and privacy issues have been vastly neglected and only a few studies [7, 8] have focused on the problems recently. In Ref. [7], the authors only focused on the security of location service and proposed a secure grid location service (SGLS) based on the GLS. In Ref. [8], only the privacy issue was considered and an anonymous location service (ALS) was proposed. But in the ALS scheme, malicious nodes may forge faulty position information. Furthermore, the computational and communication cost is too high.

In this article, both the security and privacy are considered, and a secure privacy-preserving hierarchical location service is proposed based on the original hierarchical location service (HLS) proposed by Wolfgang Kieß et al. We use a fully public key broadcast encryption scheme [9] and an efficient broadcast authentication scheme [10] to protect the position privacy and security. Our proposed scheme has a low computational and communication cost.

The remainder of the paper is organized as follows: Section 2 is the preliminaries. The proposed secure privacy-preserving location service is described in detail in section 3. Section 4 contains the security analysis and the result of simulation with ns-2. Finally the paper is concluded with a summary and an outlook to future work in section 5.

2 Preliminaries

2.1 Location Service Model and Attack Model

A location service usually consists of two components, the *location update* and the *location request.* There are three location service entities in the general location service architecture:

1) Location Updater (LU) updates the location servers with its current position information when necessary;

2) Location Server (LS) stores the position information from the LUs and replies to the location queries from the location requesters;

3) Location Requester (LR) sends location requests to the location servers and obtains the position information of the LUs.

Take HLS as example, the basic operation is as follow: the area occupied by the network is divided into a hierarchy of regions. The lowest level regions are called cells. Regions of one level are aggregated to form a region on the next higher level of the hierarchy. Regions on the same level of the hierarchy do not overlap. For any given node A, one cell in each level of the hierarchy is selected as the responsible cell by means of a hash function. As node A moves through the area covered by the network, it updates its responsible cells with its current position. When another node B needs A's position, it uses the hash function to determine these cells which may be responsible for A. It then queries those cells in the order of the hierarchy, starting with the lowest level region. On the first level that contains both nodes A and B the query will arrive at a responsible cell of A where it can be answered [6].

It is obvious that the original HLS is vulnerable to a lot of attacks:

1) Position Privacy Exposure: a) malicious nodes may request the location servers and obtain the position information of any other node in the network according to the HLS algorithm so as to track it; b) location servers will discover the cell which the LU is in when it updates its position information to the location servers;

2) Update Message Tampering Attack: malicious nodes may alter the position information of the location Updaters (LUs) in the update message;

3) Update Message Forging Attack: malicious nodes may forge faulty position information to update the location servers so as to influence the communication between other nodes;

2.2 Broadcast Encryption Scheme

In order to protect nodes' position information, a broadcast encryption scheme is used. The broadcast encryption scheme allows the sender (Location Updaters) to securely distribute its position information to a dynamically changing group of receivers (Location Requesters) it trusts over a broadcast channel. In Ref. [9], the authors proposed a fully public key broadcast encryption scheme (FBE) in which the trusted receivers can choose the secret key by themselves. Furthermore, in the scheme new trusted receiver can be added and the no longer trusted receiver can be eliminated from the receiver group easily. The FBE scheme can be formalized as the following algorithms:

1. Key Generation

The sender publishes some information I and the trusted receivers randomly choose their secret keys $SK_1, SK_2 \cdots SK_n$ (n is the number of trusted receivers) and compute the corresponding public keys $PK_1, PK_2 \cdots PK_n$. The sender collects the public key and computes the system encryption key PK.

2. Broadcast Encryption ($BEnc$)

The sender randomly chooses a session key s, encrypts it with PK to make the enabling block $\Gamma = BEnc(s, PK)$ and the cipher block $C' = E(m, s)$. Finally it broadcasts $C = <\Gamma, C'>$.

3. Conversion

On receiving the broadcast C, every receiver U_i coverts it to $C_i = Con(PK_i, C)$.

4. Decryption

U_i decrypts the enabling block in C_i with SK_i and gets the session key s and uses s to decrypt the cipher-text block and gets the message m.

5. Add User

The sender produces the new system public key PK_{new} with the old one PK and the new receiver's public key PK_j according to the key generation algorithm.

6. Revocation

The sender collects the currently trusted receivers' public key and re-computes the system public key in order to eliminate the no longer trusted receivers from the trusted group.

2.3 Broadcast Authentication Scheme

Since faulty/malicious packet injection is possible in the location service algorithm, a broadcast authentication scheme is required to enable the receivers (Location Requesters) to verify the validity of the replies from the Location Servers.

In this article, we use an efficient broadcast authentication scheme proposed by Shang-Ming Chang al. [10], which is a lightweight one-time signature scheme that allows the receivers to authenticate broadcast messages from any sender. The scheme also has no authentication latency and the overhead is negligible. So it is suitable for our secure hierarchical location service with frequent location update and request.

The broadcast authentication consists of three phases: initial phase, signing phase and verification phase. In the initial phase, the sender generates a private key K_{pri} and its corresponding public key K_{pub}. Next the sender uses the private key in the signing phase to sign a message. Finally, the receivers use the sender's public key to validate the signature of the message. The first public key K_{pub} is distributed to the receivers in the initial phase by the sender using its publicly known public key PK_i (we assume that any node in the network has a pair of asymmetric keys PK_i / SK_i). For efficiency, the sender may distribute the next public key generated in the initial phase by authenticated broadcasting using the old private key generated in the last initial phase [10].

3 The Proposed Scheme

In this section, we describe the proposed secure privacy-preserving hierarchical location service in detail. Since the proposed scheme is based on the original HLS algorithm, we will emphasize on the location update phase and position request phase which are different from the original HLS algorithm.

3.1 Area Partitioning and Responsible Cells

Structure of the hierarchy of regions in the proposed scheme SPPHLS is similar to the original HLS (illustrated in Figure 1). Area of the ad-hoc network is partitioned into cells. The shape and size of the cells can be chosen arbitrarily according to the properties of the network. The cells are grouped hierarchically into regions of different levels. Take Figure 1 as example, a number of cells form a level-1 region and a number of level-1 regions form a level-2 region and so on. Regions of the same level in the hierarchy do not overlap.

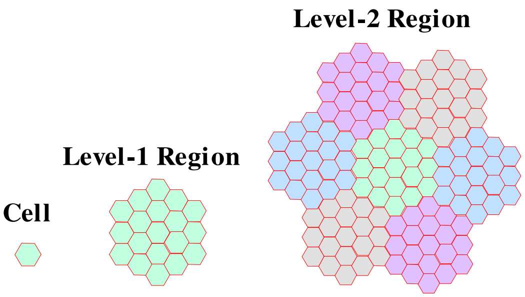


Fig. 1. Structure of Regions

The location information of a node T is placed in a set of cells called responsible cells (RCs) of T. When T sends an update packet to an arbitrary node within or close to a responsible cell R, this node becomes location server for T. Here, the routing for SPPHLS is done with a position-based routing like GPSR [1]. A node T selects one responsible cell for each level in the hierarchy through a hash function [6]. Take Figure 2 as example, R1, R2 and R3 are responsible cells of node T in the corresponding level-1 to level-3 regions according to its current position.

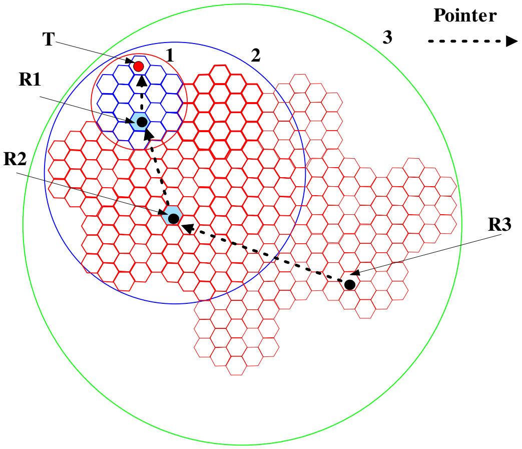


Fig. 2. Responsible Cells of Node T

3.2 Location Update and Position Request

In the original HLS, when a node updates its location servers, the servers will discover the cell which the node is currently in. As illustrated in Figure 2, when node T updates its location servers (in responsible cells R1, R2 and R3), these location servers will discover the cell which T is currently in. Thus, the position information of node T is exposed and the privacy is badly threatened. Furthermore, a malicious node may personate itself to be node T and update the corresponding location servers with faulty location information so as to influence the communication between node T and other nodes. On the other hand, any node in the network can obtains the location information of node T through the location service.

In the proposed SPPHLS, we use the broadcast encryption and broadcast authentication mechanisms to protect the security and privacy. What is more, the location update proceeding in SPPHLS is different from the original HLS. In original HLS scheme, the node which has move out of one cell will update the location server in the RC of the lowest level-1 region with a pointer pointing to the node's current cell and other location servers in the RCs of higher level regions with pointers pointing to the RC of the next lower level (like Figure 2). But in the proposed SPPHLS scheme (take Figure 3 as example), to update its location information, node T does not send the update packet to the location server in the responsible cell R1 of level-1 region so as to hide itself in a larger area (area in the blue circle in Figure 3). Instead, it updates the location server in R2 with a cipher-text including its position information (we will discuss about the cipher-text later) and other location servers of higher levels with pointers pointing to the RC in the next lower level region. Of course, if node T requires much higher level privacy, it can start the updating from the location server in RC of level-3 or more higher level regions, and ignore the location servers of RCs in lower level regions.

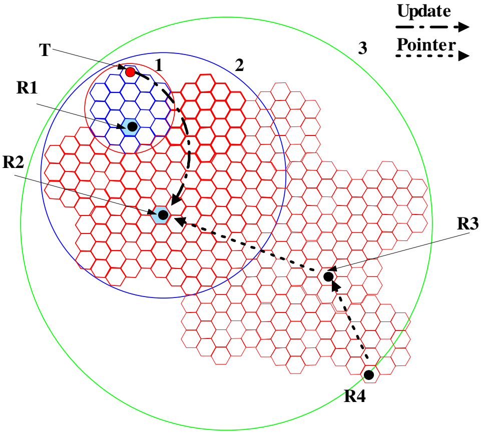


Fig. 3. Location Update

Node T generates the cipher-text of its position information with the broadcast encryption algorithm and broadcast authentication algorithm described above. The cipher-text can be denoted as $(ID_T, BA(BE(m)))$ for simple, here ID_T is the identity of node T used as the index, BA indicates the Broadcast Authentication algorithm, BE indicates the Broadcast Encryption algorithm and m is the message about node T's position. Then node T stores the cipher-text onto the location server in the RC of the certain level region which it selects as the lowest level region and updates the RCs of higher level regions with pointers pointing to the RC in the next lower region when necessary.

To successfully query the current location of a target node T, the request of a source node S needs to be routed to the location server which contains the cipher-text about node T's position information. Node S knows the structure of the candidate tree [6] defined via the hash function and T's ID. Thus the request only needs to visit each candidate cell of the regions containing S. The candidate cell of the region with the lowest level containing both S and T and the cipher-text or the pointer is per definition a responsible cell, and so are the candidate cells of all higher levels.

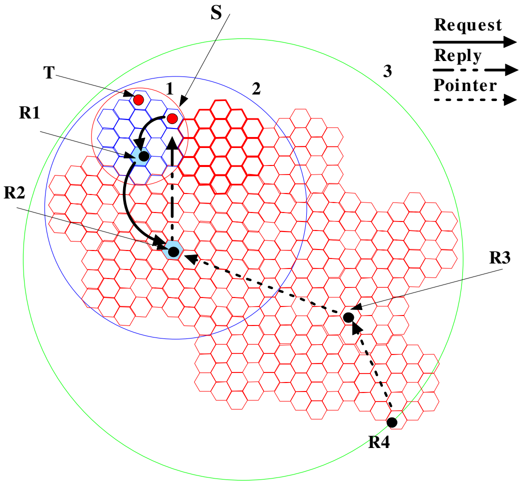


Fig. 4. Location Request

Take Figure 4 as example, node T requires the level-2 privacy, in other words, it wants to hide itself in the level-2 region (the area in the blue circle), so when updating the location servers, it will store the cipher-text about its current position onto the location server in the responsible cell R2 of level-2 region and other location servers in responsible cells of higher levels like R3 and R4 with pointers. When source node S wants to query the location of node T, it first request the level-1 responsible cell R1, though node T and node S are in the same level-1 region, R1 doesn't contains the cipher-text about node T's current position. Then the request will be forwarded to the responsible cell R2 in the level-2 region for the position information. And R2 replies to node S with the cipher-text about node T's current location information. Node S then authenticates the received cipher-text and decrypts the cipher block to get the position of node T. The authentication algorithm and decryption algorithm have been described above in Section II and can refer to Ref. [9] and Ref. [10].

Take Figure 5 as example when node S and node T are located in the same level-3 region. Node S first requests the candidate cell $R1'$ of the level-1 region containing S and there is nothing about the target node T's position information, so the request is forwarded to $R2'$ and finally reaches the level-3 candidate cell $R3$ which contains a pointer pointing to the responsible cell which stores the cipher-text about node T's

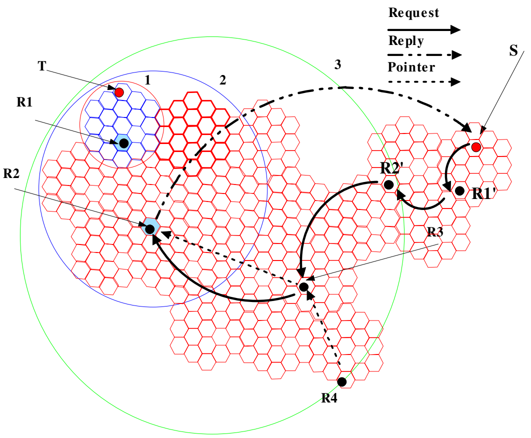


Fig. 5. Location Request

position information. Then the request is forward according to the pointer down to the level-2 responsible cell $R2$ and the location server in $R2$ reply to node S with the cipher-text about node T's location information.

4 Security Analysis and Simulation

In this section, we will analysis the security of the proposed scheme and show how the privacy is protected.

1) Against position privacy exposure

a) Malicious node can no longer get the position information of the intended node because the position information is encrypted with the broadcast encryption algorithm, only the nodes trusted by the intended node can decrypt the cipher text and get the position information with the private key SK_i generated at the Key Generation Phase of the broadcast encryption algorithm.

b) For the position information has been encrypted, location servers can no longer judge the cell in which the Location Updater is. Furthermore, Location Updaters can update the position information onto the RCs in higher level regions only and neglect the RCs in low level regions so as to hide itself in a larger area according to the level of privacy required.

2) Against Tampering and Forging Attacks

For the cipher text about the position information has been authenticated according to the broadcast authentication algorithm, so the malicious nodes can not modify or forge the update message, and the security of the location service is protected. On receiving the cipher text $(ID_T, BA(BE(m)))$, Location Requesters can verify it using the public key K_{pub} of the sender according to the broadcast authenticated algorithm. After authentication, the Location Requesters use the private key generated in the Key Generation Phase of the broadcast encryption algorithm to decrypt the cipher text $BE(m)$ and get the message m .

For the simulations, the discrete event simulator ns-2 [11] version 2.29 was used. The nodes move according to the Modified Random Direction Mobility Model [12] and the GPSR routing protocol was used. We selected the original Hierarchical Location Service (HLS) as a benchmark and compared the success rate and response time of the proposed scheme (SPPHLS) with it in the following scenario (illustrated in Table 1):

Table 1. Parameters for Simulation

Number of nodes	150
Node density per square kilometer	67
Area Size (kilometer*kilometer)	1.5*1.5
Max Speed(m/s)	10,30,50
Simulation Time(seconds)	300
Request per Node	4
Request per simulation run	600
MAC Layer	IEEE 802.11

1) Success Rate

Success Rate is the percentage of queries which have been successfully answered by the location servers. A query is answered successfully if the location service can provide position information of the target node with a precision of at least 250m. In Figure 6, SPPHLS achieves success rates of 84% to 93%. It is a few lower than the original HLS algorithm and the degradation is due to the delay of response time introduced by the proposed SPPHLS for the authentication and encryption operation, and the change of position update procedure. Compared to the original HLS algorithm with success rate of 85.5% to 93.5%, the result is acceptable.

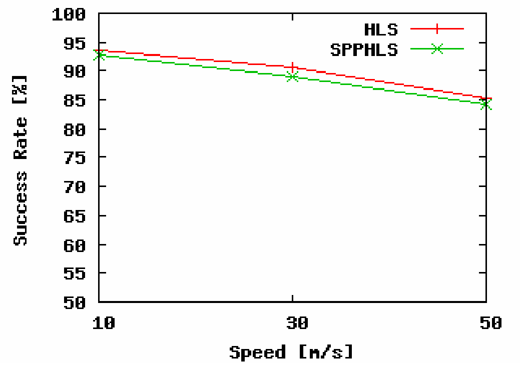
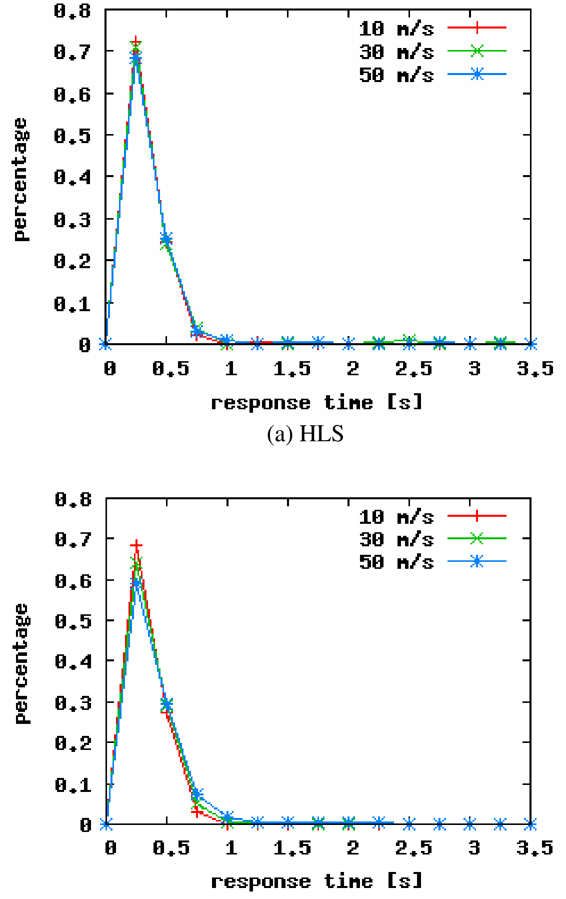


Fig. 6. Success Rate of HLS and SPPHLS

2) Response Time

Response Time is the time between sending a request and reception of a reply. Figure 7 presents the response time of the original HLS and the proposed SPPHLS in the simulation scenario. Each point at a time coordinate x in the chart stands for the interval $(x-0.25,x]$. The shapes of the curves are results of the communication load on the network. The increasing of speed causes more update messages to be sent and the higher load produces more collisions resulting in unwanted delay. Of course the authentication and encryption operation also introduce some delay.

Finally, the average response time is showed in Figure 8.The response time is composed of: (1) the time it takes a packet to be forwarded between the target candidate cells; (2) the timeout for cellcasts[6] if the cellcast failed; (3) the time it takes the reply to reach the source. In Figure 8, the average response time of SPPHLS is a little larger than the original HLS algorithm due to the authentication and encryption operations, and modification of the location update and request procedure.



(a) HLS

(b) SPPHLS

Fig. 7. Response Time

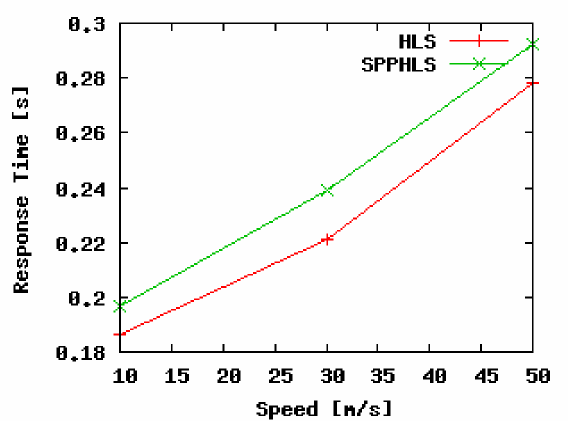


Fig. 8. Average Response Time

5 Conclusion and Outlook

In this article, we propose a secure privacy-preserving hierarchical location service for geographic routing in mobile ad hoc networks. The proposed location service scheme addresses two critical issues in geographic routing, i.e. the privacy and security issues. And the simulation shows that the proposed scheme is practical for the very moderate success rate degradation and response time it introduces. In the future, we will focus on improving the authentication and encryption algorithms used by the secure location service and developing a more efficient caching mechanism to further improve the success rate and reduce the response time.

Reference

1. Karp, B.N., Kung, H.T.: GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. In: MobiCom 2000. Proceedings of the sixth annual ACM/IEEE International Conference on Mobile computing and networking, Boston, Massachusetts, pp. 243–254 (August 2000)
2. Mauve, M., Widmer, J., Hartenstein, H.: A survey on position-based routing in mobile ad hoc networks. IEEE Network Magazine 15(6), 30–39 (2001)
3. Camp, T., Boleng, J., Wilcox, L.: Location Information Services in Mobile Ad Hoc Networks. In: Proc. of IEEE ICC 2002, New York City, New York, pp. 3318–3324 (April 2002)
4. Li, J., Jannotti, J., DeCouto, D.S.J., Karger, D.R., Morris, R.: A Scalable Location Service for Geographic Ad Hoc Routing. In: MobiCom 2000. Proceedings of the sixth annual ACM/IEEE International Conference on Mobile computing and networking, Boston, Massachusetts, pp. 120–130 (August 2000)
5. Xue, Y., Li, B., Nahrstedt, K.: A scalablelocation management scheme in mobile ad-hoc networks. In: LCN'2001. Proc. of the IEEE Conference on Local Computer Networks, Tampa, Florida (November 2001)

6. Kieβ, W., Füβler, H., Widmer, J., Mauve, M.: Hierarchical location service for mobile ad-hoc networks. ACM SIGMOBILE Mobile Computing and Communications Review 8(4), 47–58 (2004)
7. Song, J.-H., Wong, V.W.S., Leung, V.C.M.: A framework of secure location service for position-based ad hoc routing. In: Proceedings of the 1st ACM international workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks, Venezia, Italy, pp. 99–106 (2004)
8. Zhi, Z., Choong, Y.K.: Anonymizing Geographic Ad Hoc Routing for Preserving Location Privacy. In: ICDCSW 2005. Proceedings of the Third International Workshop on Mobile Distributed Computing (MDC), vol. 06 (2005)
9. Tan, Z.W., Liu, Z.J., Xiao, H.G.: A fully public key tracing and revocation scheme provably secure against adaptive adversary. Journal of Software 16(7), 1333–1343 (2005)
10. Chang, S.-M., Shieh, S., Lin, W.W., Hsieh, C.-M.: An efficient broadcast authentication scheme in wireless sensor networks. In: ASIACCS 2006, Taipei, Taiwan (March 21-24, 2006)
11. The ns-2 network simulator, http://www.isi.-edu/nsnam/ns/
12. Royer, E.M., Melliar-Smith, P.M., Moser, L.E.: An Analysis of the Optimum Node Densityfor Ad hoc Mobile Networks. In: Proceedings of the IEEE International Conference on Communications, Helsinki, Finland (June 2001)

LBKERS: A New Efficient Key Management Scheme for Wireless Sensor Networks

YingZhi Zeng, JinShu Su, Xia Yan, BaoKang Zhao, and QingYuan Huang

School of computer, National University of Defense Technology, ChangSha Hunan, China
zyz1234@gmail.com

Abstract. The framework of security communication among sensor nodes is the most important aspect and a basic research field of securing Wireless Sensor Networks. Many techniques have been developed recently to establish keys in sensor networks. Most of them are based on cluster topology. Based on the novel loop-topology, this paper proposes a new key establishment and rekeying scheme. The loop-based scheme has many advantages over cluster-based scheme. The analysis in this paper demonstrates its feasibility, efficiency and security for key establishment and maintenance in Wireless Sensor Networks.

1 Introduction

In a wireless sensor network, sensor nodes are typically deployed in adversarial environments such as military applications. Sensor nodes may be dropped from airplanes to the work places and need to communicate later with each other for data processing and routing. The unattended nature of the deployed sensor network lends itself to several attacks by the adversary, including physical destruction of sensor nodes, security attacks on the routing and data link protocols, and resource consumption attacks launched to deplete the limited energy resources of the sensor nodes. Unattended deployment also makes insider attack easier.

In a word, the wireless connectivity, the absence of physical protection, the close interaction between sensor nodes and their physical environment, and the unattended deployment of sensor nodes make them highly vulnerable to node capture as well as a wide range of network-level attacks. Moreover, the constrained energy, memory, and computational capabilities of the employed sensor nodes limit the adoption of security solutions designed for traditional networks. Thus the encrypted communication is crucial to the secure operation of sensor networks. Firstly a large number of keys need to be managed in order to encrypt and authenticate all sensitive data exchanged. Secondly the characteristics of sensor nodes and WSNs render most existing key management solutions developed for other networks infeasible for sensor networks. To provide security communication key in such a distribution environment, the well-developed public key cryptographic methods have been considered at first, but these demand excessive computation and storage from the resource extra-limited sensor nodes [1]. The symmetric key cryptography is considered as the only feasible way for wireless sensor networks. Therefore, there must be a secret key shared between a pair

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 772–783, 2007.

of communicating sensor nodes. Sensor nodes can use pre-distributed keys directly, or use keying materials to dynamically generate pair-wise keys.

The broadcast communication mode and the resource-constrained feature of sensor nodes decides that generating a unique communication key for every two nodes is costly and unpractical. Due to the high communication cost and limited bandwidth, it is not feasible either for each node to send its sensor data directly or through relay nodes to the sink. The relations of sensor data generated by neighbor nodes lead out that data aggregation is needed. There are two kinds of communication situation that we should keep them secure: the data transferred from normal nodes to the aggregator nodes; the aggregated data transferred from the aggregators to the sink.

The next important topic we should focus on is the network topology, which is related to the communication connectivity and the total coverage of target region. Since the network topology is unknown prior to deployment, the key pre-distribution scheme is put forward to provide communication keys for the data aggregating flow and relaying flow, where the keys are stored in the ROMs of sensor nodes before the deployment. After deployment, each sensor node should connect with its neighboring nodes and generate their security keys in a self-organized method.

The main contribution of our work is focused on the basic framework for the security communication in WSN(Fig.1). The propose scheme includes key material pre-distribution, key establishment, key maintenance Trigger and key rekeying. In particular, the main framework is based on a novel loop-based topology for WSN. Our analysis and comparison indicate that this scheme has substantial advantages over the traditional cluster topology schemes.

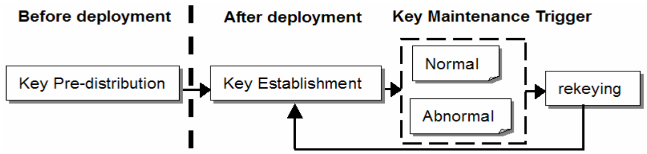


Fig. 1. The proposed key Management framework of security communication in WSN

This paper is organized as following: Section 2 describes the related works. Section 3 compares cluster topology with loop topology. The construction of the self-organized loop-topology, relative key establishment and rekeying scheme are presented in Section 4. Section 5 deals with the detailed performance analysis and comparisons. Section 6 concludes the paper with future research directions.

2 Related Works

For the sake of generating key for security communication in WSN, many Key Establishment Schemes (KES) have been proposed in recent years. Due to the obvious shortcoming, Key Distribution Center scheme and PKI-based scheme are both unfeasible in realization. Key Pre-Distribution (KPD) is the hot spot in this area. The main KPD schemes include random schemes and determinate schemes.

Eschenauer and Gligor [2] proposed the first random key pre-distribution scheme. Each sensor node is assigned k keys out of a large pool P of keys in the pre-deployment phase. Neighboring nodes may establish a secure link only if they share at least one key, which is provided with a certain probability depended on the selection of k and P. Based on the EG scheme, q-composite keys scheme was proposed by Chan in [3]. Using the framework of pre-distributing a random set of keys to each node, Chan presented other two mechanisms for key establishment: a multi-path key reinforcement scheme and a random pair-wise keys scheme.

Liu-Ning ,Du and Choi-Youn schemes can be considered as determinate schemes in which any two nodes can share a certain common key according to some mathematics rules. Liu and Ning [4] provided further enhancement by using t-degree bi-variate key polynomials. Du et al. [5] proposed a method to improve the basic scheme by exploiting a priori deployment knowledge. They also proposed a pair-wise KPD scheme for WSN[6], which uses Blom's key generation scheme [7] and basic scheme as the building blocks. Choi and Youn [8] proposed a KPD 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.

Random schemes have a common security problem: they cannot keep remain nodes being safe if some nodes are compromised and those keys loaded on the nodes are exposed to enemy. The initial key selection is random and the final key distribution is out of control. Even the exposed keys can be found out with the compromised nodes being detected, the next cleaning work is out of the question.

Determinate schemes rely on certain mathematics rules. If the number of compromised nodes is increased to a level, the enemy can use the known key material to deduce the mathematics rules for key generating.

Generated keys are distributed among sensor nodes to keep the communications secure. All of above schemes aim to find a best way to establish keys, but they ignore an important thing: nodes playing different roles should use different keys according to different situations. A node may be a normal sensor, a data aggregator or a relay node. The data flow can also be different. Which role nodes should be chosen is related to the topology of WSN directly. Most old schemes simple use cluster topology as their basic organization among sensor nodes.

3 Self-organized Topology of WSN

Section 2 proposes the shortcoming of current KES schemes. From the data-center's viewpoint, the sensor data of WSN is collected and aggregated through the collaboration among neighbor nodes. So the self-organized topology of WSN is very important in key establishment phase.

3.1 Basic Definitions of Loop

In the graph theory, a loop is a non-directional path, which begins and ends with the same node. Since there is at most one connection between every two nodes in an undirected graph G=(V, E) [9], a path from v_i to v_j representing a wireless sensor network link can be defined as a sequence of vertices $\{v_i, v_{i+1}, \ldots ,v_j \}$, where V representing the set of nodes and E is the set of connections.

Loop length: The length of a loop also can be called path length, is the number of hops from v_i to v_j. Let L be a loop. It is evident that if length (L)<3, either the node on L is isolated or L is a round trip between two nodes.

Loop type: In a large scale WSN, there may be some isolated nodes. A loop with only two nodes is also a special loop. For example, in Fig.2(b), L2 and L3 are typical loops and L1 is a two noded special loop. In the following parts, nodes on the loops with greater length than 2 are called on-loop nodes.

3.2 Basic Definitions of Cluster

In the graph theory, let $G = (V, E)$ be a connected graph and $C = \{C_1, . . . , C_k\}$ a partition of V . We call C a clustering of G and C_i a cluster; C is called trivial if either $k = 1$, or all clusters C_i contain only one element. We often identify a cluster C_i with the induced subgraph of G, i.e., the graph $G[C_i]:= (C_i, E (C_i))$, where $E (C_i) := \{\{v\} |v\in Ci\}$. Then $E(C)=\bigcup_{i=1}^{k} E(C_i)$ is the set of intra-cluster edges and $E \setminus E(\mathcal{C})$ is the set of inter-cluster edges. The set $E(C_i{}^{\text{ʿ}}C_j) := \{\{v^{\text{ʿ}}w \} \in E: v\in Ci, w\in Cj \}$is the set of edges that have one end-node in C_i and the other end-node in C_j .

In the research of sensor networks, a lot of applications use cluster as the basic organization. The feature of wireless broadcast communication introduces the concept of a cluster-header. A cluster-header is chosen among its neighbor nodes according to some rules. A node acting as a cluster-header would have the power in control of its cluster-members, such as node B, D and H in Fig2 (a).

Cluster length: In a sensor networks, cluster length is the number of nodes in a cluster. **Cluster type**: According to different cluster-header rules, there is also different cluster-header forming type: such as the lowest-ID cluster or the maximum connection-degree cluster. Neighbor clusters may share some common nodes: some cluster headers may be another cluster's members in some certain situations.

3.3 The Topology of Key Establishment in WSN

As a data-center network, the core function of WSN is aggregating data and forwarding data through some relay nodes to the sink. So we consider the key establishment topology and the data process topology should not be separated.

Old KES are mainly based on cluster topology. There exist some key establishment schemes for WSN that are based on the cluster topology [12~14]. In a WSN, every sensor node acts either as a data producer or just as a router. In cluster-topology, each node should take part in a voting to choose some nodes acting as cluster headers(maybe choose itself). After the deployment and the CH's voting, the cluster headers play an important role in the next steps which include initializing keys, distributing cluster keys and rekeying. There are two kinds of working flows in cluster-based key establishment schemes: Key work flow being under the control of those cluster headers, and sensor data aggregating flow being processed between those nodes doing sensor works. The most notable problem of cluster-based KES is the permission of nodes. If a cluster member acts as a local aggregator, then its cluster header also has to send sensor data to this member. Which one has the higher power? The header is the cluster controller that is selected from neighborhoods. The aggregator is the closest node to the sink. It is difficult to choose either of them to be the commander: from the efficient of data aggregation, a cluster header acting as an aggregator will cost more energy in data transmission; if an aggregator can command a cluster header to do something, then any compromised cluster member could act as a fake aggregator and the whole cluster would run out of control.

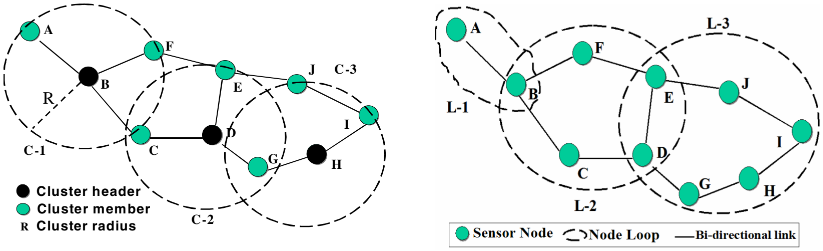


Fig. 2. (a) An example for cluster-based WSN; (b) the same example in loop-based topology

We take loop as the basic unit and the entire network is grouped into self-organized inter-connected loops. Within a loop, nodes can exchange information with each other by forwarding messages along the loop in either of the two directions. For inter-loop communications, messages are first routed to the gateways nodes (router nodes joining multiple loops) and transferred from gateway to gateway till reaching the destination. As for inner Loop transmission, messages are finally forwarded to the destination. Loop topology has many special benefits in WSN.

Efficiency: The loop topology is more adapt to the physical positions of sensor nodes. When a node within a loop receives an order from the sink to sense some special information, the node becomes an aggregator immediately. Its neighbor nodes broadcast this message to farther neighbors and act as lower level aggregators. Each level aggregator will compare and integrate data within its loop. The result would be shortened before it is sent to the higher level aggregator. The sensor data would be shortened and be aggregated level-by-level till it arrives at the sink. It is obvious that data aggregation in loop topology without the level of cluster header is more efficient and cost-saving.

Simple Structure: After the loop topology is formed, there is no critical header node defined in a loop. This simple network topology never suffers from the chain change caused by the re-election of headers. The overlay situation of neighbor clusters is complex. Two neighbor loops just share common edges at most. The relationship of neighbor loops is also very simple. The cluster length is larger than the loop length, so the number of loops in a WSN is more than the number of clusters. But the simple structure of loop-based scheme leads to simple management and collaboration for loops. On the contrary, the relationship between neighbor clusters is complex, especially during the cluster header's renewing time.

Robustness: Local loop information can be reserved in every node on the loop. This information redundancy enhances the network robustness. There exist two paths between every two nodes in the same loop. This feature provides a backup route and authentication path for link failure during the transmission of aggregated data.

4 The Proposed Key Management Scheme

4.1 Creation of a Loop Topology for Key Establishment in WSN

Key Pre-distribution: Each node should be assigned some key materials, including a unique ID, a private key, a Hash function and a global key.

Four-rounds Broadcasting: After deployment, each node starts broadcasting its ID message encrypted by the global key. This action can prevent malice listening during the initialization phase. Each node which receives a message can build up its neighbor table.

After checking their neighbors' information, each node starts second broadcasting its neighbor numbers to neighbor nodes.

There always exist some nodes with larger neighbor numbers than neighbors whose neighbors can start broadcast their neighbor tables (NT). The NT messages would be received by neighbor nodes and be added into their link tables(LT).

Those nodes with larger neighbor numbers nodes can broadcast the latest LT messages to neighboring nodes.

Some neighbor nodes may have the same neighbor numbers. For the purpose of saving cost, we set a rule: if two adjacent nodes have equal number of neighbors and this number is greater than 2, then either the node with smaller ID or the node with larger ID should broadcast his NT table first. They cannot broadcast at the same time. Here is also another **special situation** for loop that we should take into account. In the application of WSN, the time of loop converging rate is also very important. If sensor nodes are deployed sparsely that some of them may have only one neighbor, we require that those nodes with only one neighbor can send their loop constructing message in the second broadcasting round. According to the example of Fig 2 (b), the broadcasting process and sequence are shown in Figure 3.

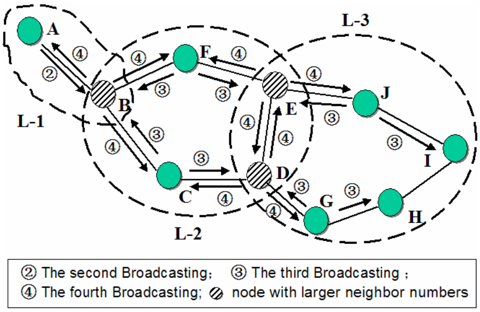

L-3
A
④
②
④
F
④
④
L-1
B
③
③
E
J
③
④
③
④
③
③
I
④
③
C
D
③
G
③
④
④
H
L-2
② The second Broadcasting;
③ The third Broadcasting ;
④ The fourth Broadcasting;
▨ node with larger neighbor numbers

Fig. 3. An example of loop creation

Constructing loops: After several units of broadcasting time, some nodes, such as node C in Figure-3, may receive link message{E,F,B} from left node B and {E, D} from right node D. A loop can be constructed by the conjunction of the two messages. The same loop {E, F, B, C, D, G} would also be formed at node E. Another loop L-3 can also be constructed by the conjunction of message{D, G, H} and {E, J, I} at node I in the same way. A special loop{A, B} is already being created at node B. Two loops may share two and even more common nodes, such as L-2 and L-3 in Figure-3.

4.2 The Loop-Based Key Establishment Scheme

The first stage of LBKERS is to construct loops as described in section 4.1, all the nodes are divided into different loops and some nodes are shared between two neighbor loops.

Based on the self-organized loop topology, the key of a loop can be created. If a node received two link messages from its two neighbor nodes and can just use the combination of the link messages to form a loop, then the node becomes a loop-creator and has the power to create a new key for those nodes in the loop. We can set up the computing formula of loop-key as following:

$$\text{Loop-key=Hash (the global key} \parallel \text{the smallest Loop member ID} \parallel \text{the largest Loop member ID).} \quad (1)$$

Only the loop-creator knows that which node's ID is the smallest and which is the largest. This design can efficiently prevent faking loop key. Because there maybe exist two or more loop-creators in constructing process of the same loop (such as node C and E in Fig.3), we set the content of above formula to keep the consistency of loop keys which are created by different creators at the same time.

After creation of keys, loop-creator will send the loop-key and the member list to its loop members. If the loop format is not special, the key messages will be sent to its two loop-neighbors at first. Each node on the loop will send the key to next node on the member list till some node receives same message from its two neighbor nodes.

After above steps, each node in WSN should belong to one loop group at least and should keep a loop-key shared with other loop members. Sensor data aggregation and communication within the loop should be encrypted with the loop key.

4.3 The Proposed Rekeying Scheme

As a resource-limited network, a WSN cannot afford changing loop-keys continuously. But there are still two situations in that rekeying is sometimes needed. We define the situations as key maintenance triggers.

Situation I: If a loop member is recognized as a defection node or the sink sends a command to get rid of a node from certain loop, the urgent work is to eliminate it form the loop's member list. First of all, such an abnormal message arrives at the closest safe loop member. The node becomes a temporary leader and will send a cleaning message to its two loop neighboring nodes. As shown in Figure 4(1), cleaning message should be sent hop-by-hop to each node on the loop except the defection node. At the same time, the leader node will random generate a new key for the loop and send this rekeying message to replace the old loop-key.

Situation II deals with normal rekeying. If a loop member is out of battery and cannot work properly any more, it should be deleted from the loop list. The loop-key that it shared with other members should also be abandoned. So the working flow in Figure 4(2) is to clean old loop-key stored on every loop member. The second step is to set up new loop-key. For the sake of saving rekeying time, the new key's creator is the loop node who has received the same cleaning messages from two neighbors.

In one word, rekeying process is very important in long-time WSN. Loop-key should be changed as quickly as possible if some defection nodes are found. At the same time, normal key updating is also a good method to keep WSN secure.

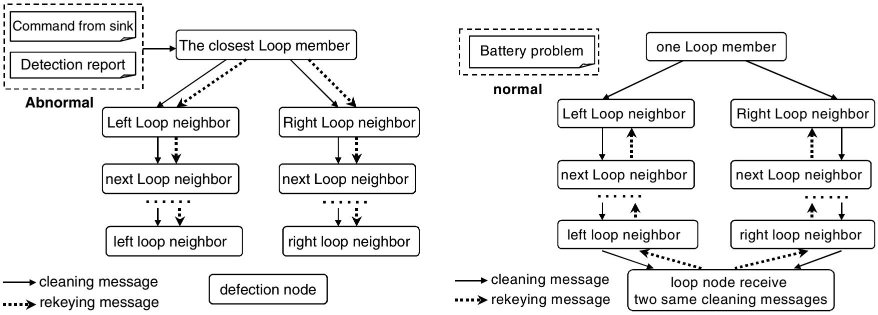


Fig. 4. Loop-based rekeying in WSN (1) Loop-based rekeying in WSN (2)

Security enhancement for rekeying: Because defection nodes can overhear neighbors' messages during the rekey procession, so some security measures should be taken to keep the communication among remain nodes of loop in the overhearing area to be safe. We assume that a defection node can only overhear his one-hop area. It is clear that we cannot prevent a defection node from overhearing the first cleaning message, but we can stop him to overhearing new keys and cause other damages. In Figure 2, if the node I is defected, link E-J and G-H should use a new key that node I cannot compute it based on the pre-shared material and overheard contents.

We use the polynomial-based key pre-distribution protocol proposed by Blundo et al.[10] to establish a new key shared between the last cleaning message's sender and receiver. The new key is only created and used between the sender and receiver, so it is a pair-wise key. Firstly before the deployment of the sensor nodes, one key sever randomly generates a bivariate t-degree polynomial $f(x, y) = \sum_{i,j=0}^{t} a_{ij} x^i y^i$ over a finite field F_q, where q is a prime number that is large enough to accommodate a cryptographic key, and has the property of f(x, y) = f(y, x). For each sensor node i with a unique ID, the key server computes a polynomial share of f(x, y), that is, f(i, y). For any two sensor nodes i and j, node i can compute the common key f(i, j) by evaluating f(i, y) at point j, and node j can compute the common key with i by evaluating f(j, y) at i. So to establish a pair-wise key, both nodes just need to evaluate the polynomial with the ID of the other node without any key negotiation and the defection nodes know nothing of the new key. The scheme is proved secure and t-collusion resistant in mathematics.

If a node finds that its loop neighbor is compromised, it can also use the special pair-wise key to encrypt the cleaning message to next-hop loop member.

Broken Loop Problem: If a loop member node has been detected as a compromised node, the loop rekeying procedure will start to process. It is obvious that the loop would be broken if one member is deleted form the loop list. The two neighbor nodes of the eliminated nodes should enlarge their communication radius to get connection to each other. By this way the broken loop can be repaired. If the eliminated node is the only gateway node between two loops, then one of the broken loop members will have to enlarge its communication radius to get connection to neighbor loops. As we know, the broken loop nodes can still collect and aggregate sense data in their work area. What we care most about is how to keep nodes working properly as long as possible.

5 Analysis and Comparison

WSNs in clustered organization are once viewed as energy-efficient and most long-lived class of sensor networks [11]. Creating a cluster for key establishment in a wireless sensor network includes at least 6 stages. In this paper we use the maximum connection-degree method as the cluster-header chosen rule.

Similar to the loop-based scheme described in Section 4.1, in cluster-based scheme, each node broadcasts its ID to neighboring nodes at first. After receiving neighbor's ID message, every node calculates its neighbor numbers and sends it to neighboring nodes. A node whose connection is bigger than its neighbors can broadcast a cluster-head-request message to its neighbors. Each node with lower connections sends a reply message to those cluster-head competitors: join or reject. Nodes that received different request messages have to choose one of those cluster-head campaigners as their cluster header. Which node to be chosen is determined by the ID or other parameters of the node. After receiving enough join messages from neighboring nodes, a cluster-head campaigner can set up a cluster key with its cluster members.

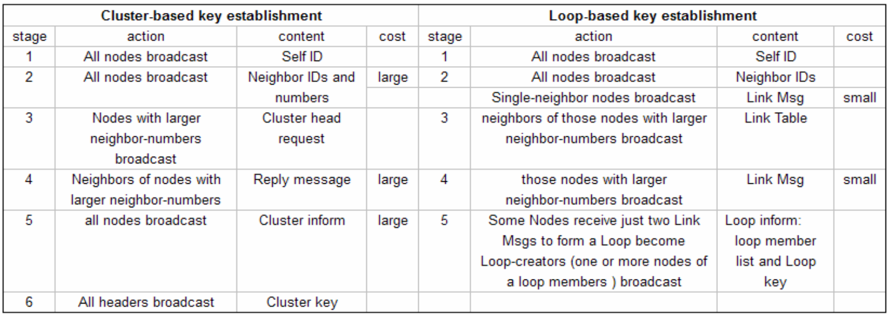
Due to the complex cluster-header chosen procedure, it is evident that the key establishment based on cluster-topology is complicated than loop-based scheme. According to the comparison in Table 1 and 2, the results are analyzed as below:

As a resource-poor network, WSN cannot afford too much **communication costs** among its nodes. The cluster-to-cluster relationship is more complex than that of loop-to-loop. It is common that some neighboring nodes are shared between two neighbor loops, but it would be redundant that more than one node are shared between two clusters: this will cost more energy in inter-clusters communication.

Communication is the biggest energy consumer. Especially the cost of sending message is much larger than receiving message. We apply CBKERS and LBKERS in the same example of Figure 2(a)-(b) and compare their sending message numbers. From Figure-5, we can find that CBKERS send more messages than LBKERS.

Storage cost: The cluster-based scheme has to save neighbor clusters' information as route in the header and some members' storage space. On the contrary, in the loop-based topology, the neighbor loop information is already broadcasted during the constructing process of the loop and only need to be stored in those nodes shared by neighbor loops.

Table 1. Cluster-based VS loop-based in communication for key Establishment



Cluster-based key establishment				Loop-based key establishment			
stage	action	content	cost	stage	action	content	cost
1	All nodes broadcast	Self ID		1	All nodes broadcast	Self ID	
2	All nodes broadcast	Neighbor IDs and numbers	large	2	All nodes broadcast	Neighbor IDs	
					Single-neighbor nodes broadcast	Link Msg	small
3	Nodes with larger neighbor-numbers broadcast	Cluster head request		3	neighbors of those nodes with larger neighbor-numbers broadcast	Link Table	
4	Neighbors of nodes with larger neighbor-numbers	Reply message	large	4	those nodes with larger neighbor-numbers broadcast	Link Msg	small
5	all nodes broadcast	Cluster inform	large	5	Some Nodes receive just two Link Msgs to form a Loop become Loop-creators (one or more nodes of a loop members) broadcast	Loop inform: loop member list and Loop key	
6	All headers broadcast	Cluster key					

As described in section 4, a cluster header acting as data aggregator may bring more communication cost to the cluster. An aggregator node needs to store and aggregate sensor data. Acting as an aggregator node at the same time, a cluster header has to afford more storage cost and computing cost for aggregating function.

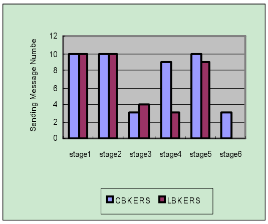
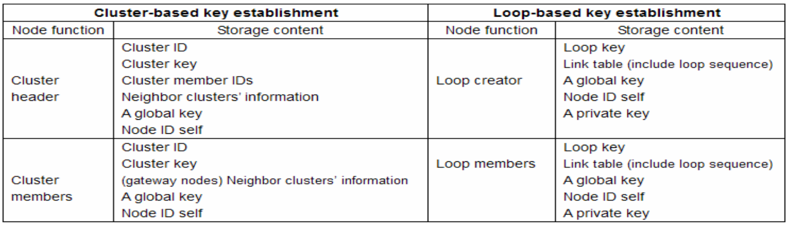


Fig. 5. Sending message numbers contrast of example in Figure-1

Table 2. Cluster-based VS loop-based in node storage for key Establishment



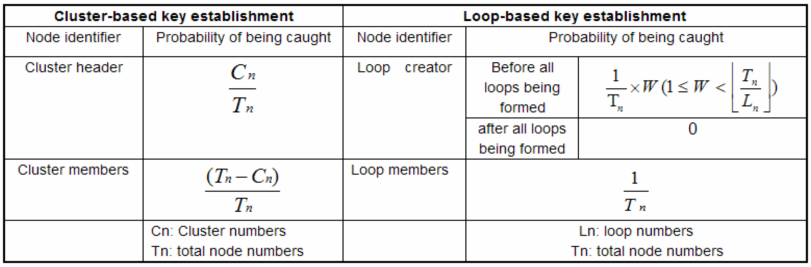
Cluster-based key establishment		Loop-based key establishment	
Node function	Storage content	Node function	Storage content
Cluster header	Cluster ID Cluster key Cluster member IDs Neighbor clusters' information A global key Node ID self	Loop creator	Loop key Link table (include loop sequence) A global key Node ID self A private key
Cluster members	Cluster ID Cluster key (gateway nodes) Neighbor clusters' information A global key Node ID self	Loop members	Loop key Link table (include loop sequence) A global key Node ID self A private key

From the perspective of **security**, the loop-based scheme is safer and more stable than the cluster-based scheme. This two schemes have different role assignment among sensor nodes. CBKERS assigns many important tasks on the cluster headers, which include: generating of Cluster ID and Cluster key; key distribution; deletion or addition node; rekeying without changing header. A header node will control its cluster members all the time until it is replaced by another node. A loop creator's identifier initializes a construction of a loop and has the right to generate a loop key. After the loop is constructed, there is no difference between normal nodes and the loop creator.

According to the probability theory, each member in a loop topology has equal probability to be caught by enemy. Once a loop member is lost, its loop-neighbors can set up new loop quickly according to the rekeying scheme: deleting the lost node ID from the loop sequence and generating a new loop key. If a cluster header is being caught, then its member nodes have to take part in a new cluster header's election. At the same time, the probability of a cluster header being caught is determined by the result that cluster numbers compare to the total node numbers. This probability is greater than that of a loop creator being caught. The probability and impact comparison results are listed in Table 3 and 4.

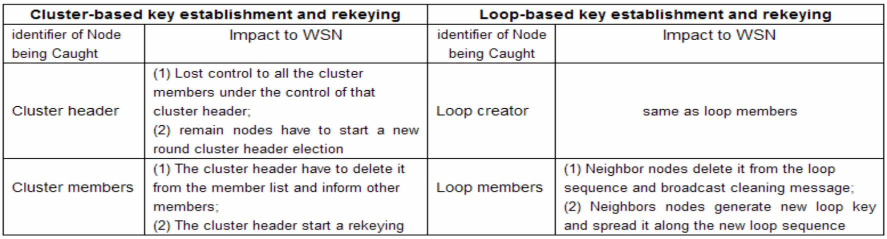
From the point of graph theory, we can also take loop as a special cluster without cluster-header. LBKERS can reduce the burden on the cluster-header and decrease relative security risk. At the same time, loop-based topology is proved to be suitable for the data aggregation in WSN. At first sensor data could be aggregated through loop-topology with the minimum cost and then the aggregated result would be transmitted through inter-loop communication to the sink. All the communication is encrypted efficiently with different loop keys. According to some special situations, rekeying scheme can be used to change loop keys timely and efficiently.

Table 3. Comparison of probability of node being caught



Cluster-based key establishment		Loop-based key establishment		
Node identifier	Probability of being caught	Node identifier	Probability of being caught	
Cluster header	$\frac{C_n}{T_n}$	Loop creator	Before all loops being formed	$\frac{1}{T_n}\times W\,(1\le W<\left\lfloor\frac{T_n}{L_n}\right\rfloor)$
			after all loops being formed	0
Cluster members	$\frac{(T_n-C_n)}{T_n}$	Loop members	$\frac{1}{T_n}$	
	Cn: Cluster numbers Tn: total node numbers		Ln: loop numbers Tn: total node numbers	

Table 4. Comparison of impact of node being caught



Cluster-based key establishment and rekeying		Loop-based key establishment and rekeying	
identifier of Node being Caught	Impact to WSN	identifier of Node being Caught	Impact to WSN
Cluster header	(1) Lost control to all the cluster members under the control of that cluster header; (2) remain nodes have to start a new round cluster header election	Loop creator	same as loop members
Cluster members	(1) The cluster header have to delete it from the member list and inform other members; (2) The cluster header start a rekeying	Loop members	(1) Neighbor nodes delete it from the loop sequence and broadcast cleaning message; (2) Neighbors nodes generate new loop key and spread it along the new loop sequence

6 Conclusion

Establishing security key is one of the most important technologies in the security mechanism of WSN. We put forward a new key establishment scheme and its rekeying scheme based on a loop topology for WSN. Differing from classic cluster topology, the construction of loop topology not only has a efficient constructing mechanism, but also has a simple and robust structure. Our scheme integrates key pre-distribution mechanism in a security framework based on the loop infrastructure. Comparing with existing cluster-based establishment schemes, LBKERS is proved to be more balanced, cost-saving, efficient and safe. Future research would focus on reduction of communication cost during the key establishment and real time detection of compromised node based on loop-topology.

Acknowledgments

This work was supported by the National Research Foundation for the Doctoral Program of Higher Education of China under grant No. 20049998027, and the National Science Foundation of China under grant No. 90104001 and No. 90604006.

References

[1] Carman, D.W., Kruus, P.S., Matt, B.J.: Constraints and approaches for distributed sensor network security. Technical Report #00-010, NAI Labs (2000)

[2] Eschenauer, L., Gligor, V.D.: A key-management scheme for distributed sensor networks. In: The 9th ACM conference on Computer and Communications, USA (November 2002)

[3] Chan, H., Perrig, A., Song, D.: Random key pre-distribution schemes for sensor networks. In: Proc. 2003 IEEE Symposium on Security and Privacy, pp. 197–213 (May 11-14, 2003)

[4] Liu, D., Ning, P.: Establishing pair-wise keys in distributed sensor networks. In: ACM Conference on Computer and Communications Security, pp. 52–61 (2003)

[5] Du, W., Deng, J., Han, Y.S.: A key management scheme for wireless sensor networks using deployment knowledge. In: INFOCOM 2004, vol. 1, pp. 586–597 (March 7-11, 2004)

[6] Du, W., Deng, J.: A Pairwise Key Pre-distribution Scheme for Wireless Sensor Networks. ACM Transactions on Information and System Security 8(2), 228–258 (2005)

[7] Blom, R.: An optimal class of symmetric key generation systems. In: Beth, T., Cot, N., Ingemarsson, I. (eds.) Advances in Cryptology. LNCS, vol. 209, pp. 335–338. Springer, Heidelberg (1985)

[8] Choi, S., Youn, H.: An Efficient Key Pre-distribution Scheme for Secure Distributed Sensor Networks. In: Enokido, T., Yan, L., Xiao, B., Kim, D., Dai, Y., Yang, L.T. (eds.) EUC 2005 Workshops. LNCS, vol. 3823, pp. 1088–1097. Springer, Heidelberg (2005)

[9] Li, Y., Wang, X., Baueregger, F., Xue, X., Toh, C.K.: Loop-Based Topology Maintenance in Wireless Sensor Networks. In: ICCNMC (2005)

[10] Blundo, C., Santix, A D, Herzberg, A., Kutten, S., Vaccaro, U., Yung, M.: Perfectly-secure key distribution for dynamic conferences. In: Brickell, E.F. (ed.) CRYPTO 1992. LNCS, vol. 740, pp. 471–486. Springer, Heidelberg (1993)

[11] Vlajic, N., Xia, D.: Wireless Sensor Networks: To Cluster or Not To Cluster? In: IEEE International Symposium on WoWMoM 2006, Niagara-Falls, Buffalo-NY, USA (June 2006)

[12] Chorzempa, M., Park, J.-M., Eltoweissy, M.: SECK: survivable and efficient clustered keying for wireless sensor networks. In: IPCCC (2005)

[13] Younis, M.F., Ghumman, K., Eltoweissy, M.: Location-Aware Combinatorial Key Management Scheme for Clustered Sensor Networks. IEEE Transactions on Parallel and Distributed Systems 17(8), 865–882 (2006)

[14] Lin, L., Ru-chuan, W., Bo, J., Hai-ping, H.: Research of Layer-Cluster Key Management Scheme on Wireless Sensor Networks. Journal of Electronics & Information Technology 28(12) (December 2006)

SAPC: A Secure Aggregation Protocol for Cluster-Based Wireless Sensor Networks

Chakib Bekara, Maryline Laurent-Maknavicius, and Kheira Bekara

Institut National des Télécommunications d'Evry
9 rue Charles Fourier, 91000 Evry Cedex, France
{chakib.bekara,maryline.maknavicius,kheira.bekara}@int-edu.eu

Abstract. To increase the lifespan of wireless sensor networks (WSN) and preserve the energy of sensors, data aggregation techniques are usually used. Aggregation can be seen as the process by which data sent from sensors to the BS are little-by-little processed by some nodes called aggregator nodes. Aggregators collect data from surrounding nodes and produce a small sized output, thus preventing that all nodes in the network send their data to the BS. Security plays a major role in data aggregation process, especially that aggregators are more attractive for attackers than normal nodes, where compromising few aggregators can significantly affect the final result of aggregation. In this paper, we present SAPC, a secure aggregation protocol for cluster-based WSN, which does not rely on trusted aggregator nodes and thus is immune to aggregators compromising. In addition to security performance, SAPC has a low transmission overhead.

Keywords: WSN, Cluster-based WSN, Data Aggregation, Security.

1 Introduction

Sensors on a WSN are generally supplied with non-rechargeable and non replaceable batteries [1]. For such sensors, transmitting is much more energy consuming than computing [2] [3]. For instance, transmitting one bit consumes as much energy as performing one thousand CPU cycle operations [2]. As a consequence, the amount of transmitted data must be reduced, in order to extend the lifetime of the network.

Aggregation techniques are usually used to reduce the transmission overhead in the network. Aggregation can be seen as the process by which data, during their forwarding from sensors to the BS, are little-by-little merged by sensors called aggregators, to produce smaller output data. The aggregation processing varies from the simple elimination of duplicated data, to the compression of data to smaller size, and mathematical operations over sensed data, like sum, average, min, max, etc. Aggregation aims to reduce the transmission overhead in the network, and consequently to reduce the sensors energy consumption.

Several aggregation protocols were introduced for WSN [5] [6] [7] [8]. However, if the aggregation process is not secured, it can be an easy target for

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 784–798, 2007.

attackers. For instance, an attacker can inject false data or modify transmitted data, or more dangerously compromise or claim to be an aggregator, in order to significantly falsify the result of aggregation. The main objective of attacking aggregation process is to produce false aggregation results, and make the BS or the network operator accept false aggregation results, so the wrong decisions and actions are taken.

To defeat attacks against aggregation process, several secure aggregation protocols were proposed in the literature [9] [10] [11]. However, these protocols either introduce some heavy communication or computation overheads [11], handle a special kind of aggregation [11], provide a limited resilience against aggregator nodes compromising [9], or require expensive interactive verifications between the BS and aggregators [10].

In this paper we present SAPC, a new secure aggregation protocol for cluster-based WSN, which does not require trusted aggregator nodes. Our protocol is resilient to nodes compromising including aggregator nodes, and introduces an acceptable transmission overhead. Our protocol allows the BS to verify the authenticity and the validity of the aggregation results, even if all aggregator nodes and part of the sensors are compromised in the network.

The rest of the paper is organized as follows. In section 2 we review some secure aggregation protocols with their performances. Section 3 presents our network model, assumptions, security goals and defines our attacker model. Section 4 details our secure aggregation protocol. Section 5 gives a detailed security analysis of our protocol and section 6 its communication overhead. Section 7 compares our protocol with some other protocols in the literature. Section 8 gives the limits of our protocol, and section 9 concludes our work.

2 Related Works

2.1 Przydatek et al. Protocol

In [10], Przydatek et al. present a secure information aggregation protocol for WSN. The authors present an aggregate-commitment-prove technique to defeat, with high probability, any attempt of an attacker to falsify the aggregation result by compromising aggregator nodes and part of sensors in the network. In addition, the authors present secure methods for computing the median and the average of measurements, finding the minimum/maximum of measurements, and estimating the size of the network. The protocol guarantees that, in the presence of a malicious aggregator, an accepted aggregation result is within an ε-error bound from the true aggregation result, where epsilon is a verifier parameter. The authors mainly describe their protocol in the presence of one powerful aggregator node in the network. Each sensor in the network shares a secret key with the aggregator node and the network operator which is in a remote location. The aggregation process is done in two steps: aggregation and commitment steps.

In the aggregation step, each sensor sends its authenticated reading to the aggregator. The reading is authenticated with two computed MACs: one generated using the secret key the sensor shares with the aggregator, and the other one using the secret key the sensor shares with the network operator. Then, the aggregator computes the aggregation result over the n data (measurements) sent by sensors, where n is the size of the network. In the commitment step, the aggregator computes the commitment tree of the sensed data, by computing a binary Merkle hash tree of depth $\log_2 n$. Each leaf of the tree represents the data sent by each node authenticated using the key the node shares with the network operator, the value of each internal node is the hash of the concatenation of its two children nodes, and the root of the tree is the commitment value. The commitment value allows the network operator to verify if the aggregation result was computed over the data generated by the sensors. Finally, the aggregator sends to the network operator the aggregation result along with the commitment value. Note that depending on the aggregation operation, the aggregator will compute one or more commitment trees.

Once the network operator receives the result of aggregation and the commitment value, it initiates an interactive verification phase to verify that the aggregator was honest, and ensures that the aggregation result is close to the true result within ε-error bound. To do that, the network operator requests the aggregator to return $\beta << n$ leaf nodes of the commitment Merkle tree, along with their corresponding path in the tree. For each leaf node, the network operator verifies its authenticity using the secret key it shares with the corresponding node, then it verifies that the corresponding path is consistent by checking if the computed root value is equal to the commitment value. If all the β paths are consistent, the network operator accepts the aggregation result and is ensured, with high probability, that the aggregator is honest.

It's obvious that if there is only one aggregator node in the network, sensors surrounding the aggregator will early deplete their energy on forwarding the readings of sensors that are far from the aggregator. Even if there are several aggregators in the network, assuming that they are powerful is not always true, because in many scenarios aggregators are selected amongst simple sensors which are extremely resource-limited devices. For such sensors, performing interactive verification-proof with the network operator is highly energy consuming, especially if they directly communicate with the network operator using one-hop communications.

2.2 Hu et al. Protocol

In [9], Hu et al. present a secure aggregation protocol, that is resilient to a single node compromising. In [9], nodes self-organize in a binary aggregation tree where only internal nodes, including the root node, are responsible of aggregation, while leaf nodes are responsible of sensing activities (see Fig.1). The protocol evolves in two steps: delayed aggregation and delayed authentication. The aggregation process is done in a delayed way, as follows:

1. Each leaf node N sends to its parent a message containing its identifier and its reading. The message is authenticated using a secret key K_N known at this instant only to N and the BS.

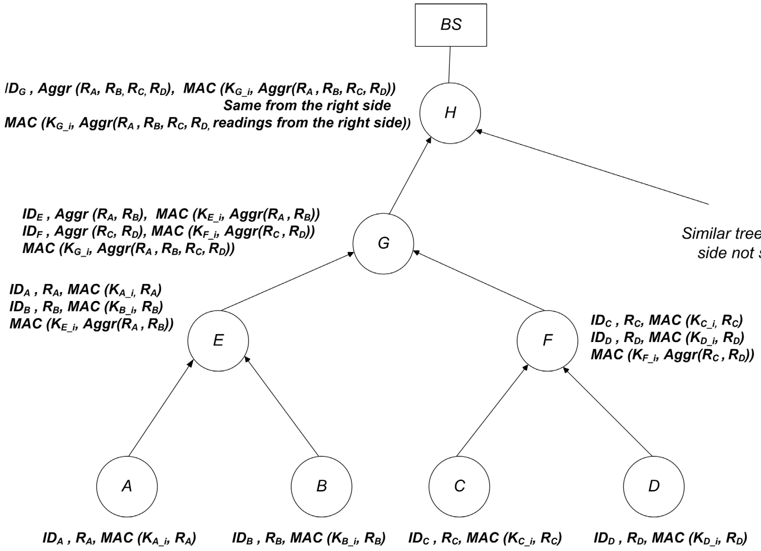

BS
H
ID_G , $Aggr(R_A, R_B, R_C, R_D)$, $MAC(K_{G\_i}, Aggr(R_A, R_B, R_C, R_D))$
Same from the right side
$MAC(K_{G\_i}, Aggr(R_A, R_B, R_C, R_D,$ readings from the right side$))$
Similar tree on right side not shown
G
ID_E , $Aggr(R_A, R_B)$, $MAC(K_{E\_i}, Aggr(R_A, R_B))$
ID_F , $Aggr(R_C, R_D)$, $MAC(K_{F\_i}, Aggr(R_C, R_D))$
$MAC(K_{G\_i}, Aggr(R_A, R_B, R_C, R_D))$
E
ID_A , R_A, $MAC(K_{A\_i}, R_A)$
ID_B , R_B, $MAC(K_{B\_i}, R_B)$
$MAC(K_{E\_i}, Aggr(R_A, R_B))$
F
ID_C , R_C, $MAC(K_{C\_i}, R_C)$
ID_D , R_D, $MAC(K_{D\_i}, R_D)$
$MAC(K_{F\_i}, Aggr(R_C, R_D))$
A
B
C
D
ID_A , R_A, $MAC(K_{A\_i}, R_A)$
ID_B , R_B, $MAC(K_{B\_i}, R_B)$
ID_C , R_C, $MAC(K_{C\_i}, R_C)$
ID_D , R_D, $MAC(K_{D\_i}, R_D)$

Fig. 1. Hu et al tree-based aggregation protocol

2. Each internal node X of level k in the aggregation tree receives a message from each of its two children Y and Z of level $k+1$ (level 0 is the tree root), and stores the messages. Using the data of the received messages, node X computes the aggregation result of the subtree rooted at its right child Y called AGR_Y, and computes the aggregation result of the subtree routed at its left child Z called AGR_Z. The aggregation result of each subtree is the aggregation of the data generated by the leaf nodes of that subtree. Then, X computes a MAC over the aggregation result of its own subtree ($f(AGR_Y, AGR_Z)$, f being the aggregation function) called MAC_X, using a secret key K_X known to itself and to the BS. Finally, X sends a message to its parent node, containing the two computed partial aggregation values AGR_Y, AGR_Z along with their corresponding MACs MAC_Y and MAC_Z respectively which are contained in the received messages, in addition to the MAC it computes MAC_X. Note that X does not send the aggregation result of its own subtree ($f(AGR_Y, AGR_Z)$), which will be computed by the parent node.
3. The process described in (2) is recursively repeated upstream until reaching the root of the aggregation tree R. In the same way, R computes the aggregation result of its right subtree AGR_{right} and left subtree AGR_{right}, in addition to a MAC MAC_R over the final aggregation result of the network using a secret key K_R known to itself and the BS. Then, R sends AGR_{right} and AGR_{left} along with their corresponding MACs generated respectively by R's right children and R's left children, in addition to the computed MAC MAC_X to the BS.

4. Upon reception of R's message, the BS discloses all the previously used authentication keys K_W in order to allow each aggregator to authenticate the stored messages (delayed authentication) sent by its children and grand-children, and thus detect any cheating aggregator node.

The protocol is resilient only to one aggregator node compromising. Indeed, two consecutive compromised nodes in the aggregation hierarchy, can collude to falsify the final aggregation result without being detected in the delayed authentication phase. This leads attackers to target aggregators in the higher hierarchy to significantly disrupt the aggregtion process, and thus produce false aggregation result. The protocol introduces a heavy communication overhead both during the aggregation phase, and during the key disclosure phase. Indeed, during the aggregation phase, data of level k are not aggregated by nodes of level $k-1$, but are aggregated by nodes of level $k-2$, resulting in extra transmission overhead. In addition, during the delayed authentication phase, the BS widely discloses n keys in the network after each aggregation round, where each key is 8-byte length, and nodes of the aggregation tree must forward the keys in reverse path, thus resulting on an additional energy consumption.

3 Background

3.1 Assumptions and Network Model

First, we suppose that the BS is a widely trusted and powerful entity, which can not be compromised.

Second, we assume static WSN, where nodes are immobile once deployed, and where nodes additions are rare.

Third, once nodes are deployed, they self-organize into clusters to save transmission energy. Different cluster formation protocols were proposed in the literature [16] [17] [18] [19], where sensors self-organize into clusters, and where routing and aggregation are performed by the cluster-heads (CHs). Our protocol uses Sun et al. protocol [19] as the underlying cluster formation protocol, where the resulting clusters form disjoint cliques, and inside each cluster (clique) each node is one-hop away from the remaining nodes of the cluster (see Fig.2). Once clusters are formed, nodes inside each cluster elect one of them to act as the CH and as the aggregator. Each CH sends to the BS the list of sensors of its cluster. For routing purpose, we suppose that the set of CHs self-organize into multi-hop routing backbone, so that CHs far from the BS can reach the BS with the minimum spent energy and receive BS's requests. Note that the result of aggregation of each cluster is sent to the BS, without being aggregated again by other aggregators.

Fourth, we suppose that each node shares a secret key with the BS, initially loaded before deployment. In addition, sensors use some key establishment mechanisms, such as [12] [13] [14] [15] for establishing secret pair-wise keys with their neighbors. To do so, each sensor is initially loaded before its deployment with some secret key materials, like a secret polynomial share [12], a secret line of a secret matrix [13] or a set of secret keys [14] [15].

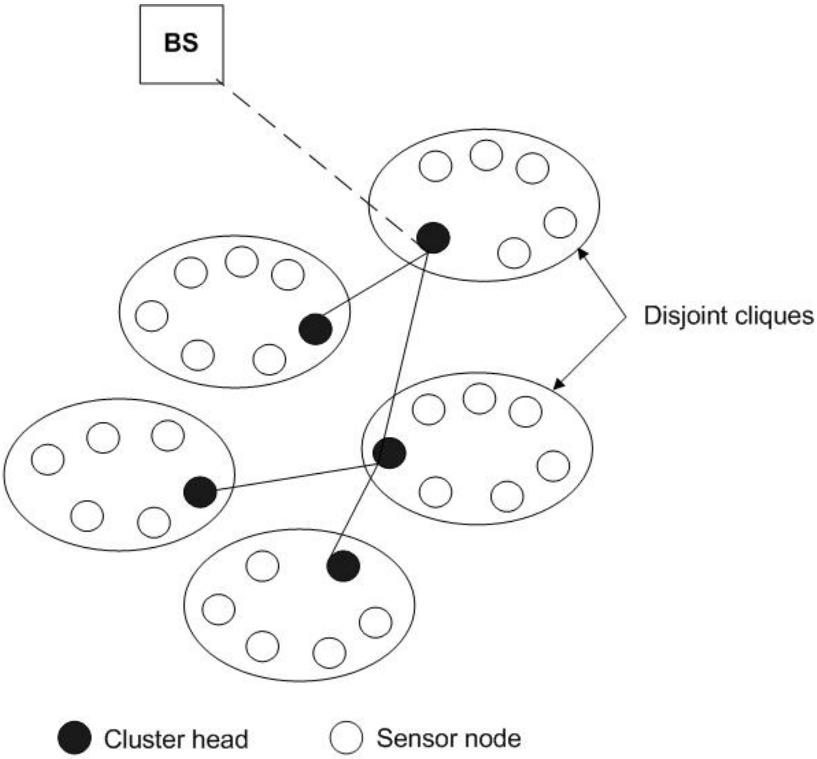


Fig. 2. Our cluster-based WSN

3.2 Adversary Model and Security Objective

We assume that an adversary can compromise a (small) portion of sensors of the network including all aggregators (cluster-heads). The objective of an attacker is to falsify the result of aggregation generated by each cluster, and to make the BS accepting false aggregation results. The easiest way for an attacker to achieve this attack, would be compromising the aggregator node and then generating an arbitrary result. The other more difficult solution would be compromising a significant portion of sensors of a cluster in order to generate a sufficient amount of bogus readings, thus generating an aggregation result which differs from the true one even if the aggregator node was well-behaving. As a consequence, it's obvious that aggregator nodes are more attractive for compromising than ordinary nodes.

Our main security objective is to protect aggregation against the compromising of aggregators, since aggregator nodes are the basis cornerstone of the aggregation process, and thus represent an ideal target for attackers to falsify the result of aggregation with the minimum effort. Our protocol does not cope with the detection of false readings reported by non-detected compromised nodes, otherwise, this would require some extra protection mechanisms like monitoring nodes behavior, or a majority-based voting mechanism like in [11].

Our protocol ensures the BS that a resulted aggregate value was computed over the original data generated by authorized well-behaving sensors of a cluster, even in the presence of compromised aggregators. So, any attempt of a compromised aggregator node to falsify the result of aggregation, either by modifying readings of well-behaving nodes, or discarding some of them will be detected at the BS.

3.3 Notations

For clarity, the symbols and notations used throughout the paper are listed in Table 1.

Table 1. Our notations

Notation	Significance
u	A sensor node
CH	A cluster-head
C_{CH_i}	A cluster headed by a cluster-head CH_i
k	The average size of a cluster
Id_u	A unique 4-byte identifier of a sensor node u
$K_{BS,u}$	An 8-byte secret key shared between node u and the BS.
C_u	A counter shared between the BS and u to prevent replay attacks
$K_{u,v}$	An 8-byte secret pair-wise key established between nodes u and v
$\{K_u^n\}$	A one way key-chain of length $n+1$ elements generated by node u
K_u^i	The i^{th} key on the key-chain of node u where $K_u^{i-1} = H(K_u^i)$, i=$1...n$
K_u^0	The commitment key of the key-chain generated by u
R_u	An 8-byte reading (measurement) generated by node u
$MAC_K(M)$	An 8-byte message authentication code generated over M using key K
H	A one way hash function, with an output length of 8 bytes
$a\|\|b$	a concatenated to b

4 SAPC: Our Proposed Secure Aggregation Protocol

As specified in 3.1, our protocol uses *Sun et al.* protocol as the underlying clusters (cliques) formation protocol. Further details on how clusters are formed are available in [19]. Our protocol evolves in three phases: *initialization*, *cluster formation* and *aggregation*. The initialization phase occurs before nodes deployment, in which the BS loads each sensor with the necessary secret cryptographic materials. Cluster formation phase occurs when nodes are deployed, in which sensors self-organize into disjoint cliques, and nodes inside each clique elect one of them as the cluster head and aggregator. Aggregation phase occurs after clusters formation, in which the aggregation process is done.

4.1 Initialization Phase

The base station loads each node u with a unique identifier Id_u, and a unique secret key $K_{BS,u}$ it shares with it. In addition, it loads u with the necessary secret cryptographic materials, that u uses in-order to establish secret pair-wise keys with its neighbors.

4.2 Cluster Formation Phase

Initially, when nodes are deployed, each node establishes pair-wise keys with its one-hop neighbors using the loaded cryptographic materials [12] and [13] [14] [15]. A pair-wise key K_{uv} is mainly used for authenticating exchanged packets between u and v, and optionally for encryption purpose. In addition, each node u generates a one-way key chain $\{K_u^n\}$ [3] to authenticate its locally broadcasted messages, and sends the commitment key of the key-chain K_u^0 to each neighbor, authenticated with the already established pair-wise key. After that, nodes self organize into clusters according to the protocol described in [19].

Once clusters are formed, nodes inside each cluster elect one of them to act as the cluster-head (CH). Each CH sends to the BS a message containing the list of sensors in its cluster. Note that in our protocol clusters are first formed, and then CHs are elected. As a consequence, in our protocol a periodic CH election inside a cluster does not change the cluster sensor members, so there is no extra overhead. In other cluster formation protocols like LEACH [16] TEEN [17] and APTEEN [18], cluster-heads are first elected then clusters are formed centered around the clusters. Thus, a periodic cluster-head election implies new formed clusters, and consequently extra energy consummation due to the exchanged messages.

4.3 Aggregation Phase

In our protocol, no trust is supposed in CHs, which play the role of aggregators. To alleviate the need of trusting aggregators, we adopt a slightly different aggregation approach than classical aggregation protocols. Instead of computing and authenticating the aggregation result by the CH only, all nodes of a cluster participate to those procedures. The BS that knows the list of sensors per cluster, can easily check whether the aggregation result of a cluster was approved by the cluster members or not, and thus knows whether the aggregator was honest or not. Aggregation process can be done either periodically, or as a response to a BS request.

In our protocol, the l^{th} aggregation round on a cluster C_{CH_i} is done as follows:

1. Each node $u \in C_{CH_i}$, including CH_i, broadcasts its reading R_u, authenticated with the current key K_u^j of its key chain:

$$u \rightarrow * : R_u \| MAC_{K_u^j}(R_u) \| K_u^j$$

2. Each node $v \in C_{CH_i}$, receives all the broadcasted messages. For each received message, node v first authenticates the disclosed key K_u^j using the stored previously disclosed key K_u^{j-1}, by checking that $K_u^{j-1} = H(K_u^j)$. Second, it verifies that the received MAC matches the message. If so, it accepts the message and replaces the stored key with the new disclosed one. The

previously stored key will be no longer used. Note that an attacker can not impersonate a node u. Indeed, communications inside a cluster are one-hop only. As a consequence, the time needed for an attacker to intercept a broadcast message sent by u and then modify the reading value R_u and generate a new MAC value using the disclosed key K_u^j, is greater than the time needed for the message to reach all nodes of the cluster. Each key is used only once for authentication, and as such each node of the cluster will only accept the first message authenticated with K_u^j, and thus will reject any further messages authenticated with K_u^j.

After collecting all readings from the cluster, each node locally applies the aggregation function over the readings to produce the resulted aggregate value: $ARG_v = f(R_u/u \in C_{CH_i})$. If we suppose the aggregation function is the sum of readings, each node $v \in C_{CH_i}$ computes:

$$AGR_v = \sum_{u \in C_{CH_i}} R_u$$

Then, each node v computes a MAC over the concatenation of AGR_v and the current counter value C_v, using $K_{BS,v}$. Then, node v sends the following authenticated message to its CH:

$$v \to CH_i : \overbrace{H(AGR_v)\|MAC_{K_{BS,v}}(AGR_v, C_v)}^{3}\|MAC_{K_{CH_i,v}} \tag{3}$$

Including C_v into the MAC computation, protects the BS from replay attacks. CH_i can also self-protect against replay attacks, by requiring that the second MAC being computed over the sequence number of each packet sent from a node of the cluster to the cluster-head CH_i

3. CH_i verifies the received messages, using the secret pair-wise keys established with nodes of the cluster. Classically, all nodes must report the same hash of the aggregate value, because all nodes of the cluster view the same broadcasted messages, and so compute the same aggregation value. Finally, CH_i computes an XOR-ed MAC over the MACs generated by nodes of the cluster over the resulted aggregate value, and sends the following message to the BS:

$$CH_i \to BS : \overbrace{AGR\| \bigoplus_{v \in C_{CH_i}} MAC_{K_{BS,v}}(AGR_v, C_v)}^{4}\|MAC_{K_{BS,CH_i}} \tag{4}$$

The message can be sent directly if the BS is in the transmission range of CH_i, or through a path constituted of other cluster-heads if CH_i is far away from the BS.

If a node $v \in C_{CH_i}$ fails to send its message, CH_i includes Id_v in the message sent to the BS, to notify that the computed XOR-ed MAC was not computed over the contribution of node v. In case of conflicting hash aggregate values (so conflicting aggregate values), CH_i can choose a majority voted hash aggregate value, and computes the XOR-ed MAC only over the MACs related to the majority voted hash aggregate value. In this case, CH_i must also report the Id of each node whose computed hash aggregate value differs from the majority voted hash aggregate value.

4. Upon receiving the message sent by CH_i, the BS verifies its authenticity using K_{BS,CH_i}. If authenticated, the BS computes a set of MACs over the received aggregate value AGR, using the set of secret keys it shares with the nodes of the cluster C_{CH_i}. The BS then, calculates an XOR-ed MAC over the computed MACs, and then compares the computed XOR-ed MAC with the received XOR-ed MAC. If the two XOR-ed MACs are equal, the BS is ensured that AGR value was computed over the original readings generated by the authorized set of sensors on the cluster, otherwise it simply rejects the MAC. It may happen that the received XOR-ed MAC is not computed over all MACs generated by nodes of a cluster, either because some nodes fail to report their result to the cluster-head or some nodes have conflicting aggregate results. Depending on the BS's policy, the BS can accept or deny the received aggregate value. If the BS has defined a threshold parameter t, the BS accepts the received aggregate result AGR, if and only if the received XOR-ed MAC was computed over at least t generated MACs, which means that at least t nodes of the cluster must agree on the same aggregate value result.

5 Security Analysis of Our Protocol

As specified in 3.2, our protocol aims to protect the BS from accepting false aggregate results, generated by a compromised, a malicious or a malfunctioning CH. By distributing the task of aggregation over all nodes of a cluster, we alleviate the need to trust a central aggregator. In our protocol all nodes participate in the computation and the authentication of the resulted aggregate value. As a consequence, a malicious or compromised CH cannot convince the BS of the validity of a false aggregate value it generates, because it cannot compute the MACs of well-behaving non-compromised sensors over the false aggregate value. Depending on the BS's security policy, an attacker has to compromise the entire cluster or part of it in order to make a BS accept a false aggregate result. If the BS requires that the computed aggregate value is computed over all the readings of nodes of a cluster, an attacker must compromise all nodes of the cluster, including the CH, in order that its attack succeeds. If the BS's policy is less strict, it can require that the aggregate value being computed over at least t readings generated by t sensors of the cluster, where $t < k$, the size of a cluster. In this case, an attacker must compromise the CH, plus *t-1* sensors of the cluster in order to make its attack possible. In general, the threshold value must be set at least to $t = \frac{k}{2}$ nodes.

Concerning the security of the aggregation process itself, each broadcasted reading R_u is authenticated using node u's current key K_u^i from its key chain.

Each key is used only once for authenticating one transmitted reading, and to authenticate the next disclosed key. As a consequence, an attacker cannot masquerade the identity of a non-compromised node by sending readings on behalf of it. The only malicious attack that remains possible is manipulating the readings of compromised nodes.

As stated in 3.2 above, our protocol is not intended to protect the network from bogus readings reported by non-detected compromised nodes or malfunctioning nodes, which can still legitimately authenticate their readings. However, this can be done either by monitoring the readings periodically sent by sensors, or by using a majority voting system. In the monitoring solution, each node monitors the evolution of readings of the nodes of a cluster. If the readings of a node are detected to be significantly different between two successive aggregation rounds, nodes of a cluster can decide to not take the reading of that node into the computation of the aggregate result. In majority-voting solutions, sensors are assumed densely deployed, so that sensors in each cluster practically report the same value of readings. In this case, the result of aggregation on each cluster is the majority voted value, like in [11]. In this way, each sensor reports as aggregate value the majority voted value, and thus malicious readings are discarded.

6 Transmission Overhead

Aggregation protocols were mainly proposed in order to reduce the amount of data transmitted in a WSN, but securing the aggregation process will certainly have some extra computation and transmission overhead. As referenced in different works [2] [3], transmitting is more energy consuming than computing. As a consequence, any proposed protocol for WSN must introduce the lowest transmission overhead as possible.

Our secure aggregation protocol attempts to introduce a small transmission overhead, while providing maximum security level. If we consider the aggregation operation is the sum of sensors readings, where each reading and cryptographic key are 8-byte length, and that an authenticated packet contains a data payload of at most 24 bytes, a header of 12 bytes (source and destination addresses, plus a sequence number), and a generated MAC of 8 bytes, the following transmission overhead applies to a cluster for each aggregation operation:

- Each node in the cluster (except the CH) broadcasts in the cluster one packet of 16-byte payload, and sends one unicast packet to its CH (the computed MAC over the aggregate value) of 16 bytes payload.
- The CH broadcasts in the cluster one packet (its reading) of 16-byte payload, and sends one unicast packet (the final aggregate value) to the BS of $16 + \frac{1}{8} \times \lg_2(k)$ byte payload. In addition, each CH which is in the path from a far CH to the BS, will forward one or more packets (aggregation results of distant clusters) of $16 + \frac{1}{8} \times \lg_2(k)$ bytes payload.

7 Comparison with Previous Works

Our comparison will mainly focus on the resilience to aggregator nodes compromising, and on the energy consumption due to the transmission overhead.

7.1 Resilience to Aggregator Compromising

Our secure aggregation protocol is resilient to the compromising of all aggregator nodes, and part of sensor nodes compromising depending on the BS's security policy (see Section 5). Hu et al. protocol is resilient to a single aggregator node compromising only, because two consecutive compromised aggregators can collude to produce a false aggregation result, that the BS will accept as valid. Przydatek et al. protocol has a high probability of detecting misbehaving and compromised aggregator nodes that report aggregation results not within the ε-error bound.

7.2 Transmission Overhead

In Hu et al. protocol aggregation is done in a delayed way, and half of sensors in the network are aggregators. This implies more energy consummation, because data sent by aggregator nodes of level k in the aggregation tree must be propagated for two-hop before being aggregated by aggregator nodes of level $k-2$. As a consequence, each internal aggregator node will forward the messages sent by its left children and right children, in addition to the MAC it computes. If we consider the aggregation function is the maximum of readings, where each reading and MAC are 8-byte length, each aggregator needs to forward at least 40 bytes. Moreover, during the delayed authentication phase, the BS widely diffuses in the network n cryptographic keys of 8-byte length each, where n is the network size. The keys are forwarded by the aggregator nodes in a reverse path. This result in expensive extra communication overhead.

In Przydatek et al. Protocol, transmission overhead is mainly due to the expensive interactive verification phase, where each aggregator node in the network must provide a proof of the correctness of its aggregation result to the network operator. The proof is a set of β leaf nodes values along with their corresponding path values in the commitment tree, where each value is at least 8-byte length. The length of the path is logarithmic to the number of sensors served by each aggregator. If an aggregator serves 32 sensors, a path in the commitment tree contains 6 hash values, so the length of a path is 48 bytes, and the total amount of data an aggregator sends back to the network operator is $48 \times \beta$ bytes. Depending on the desired aggregation function, we can have different values of β, with β is proportionally related to $\frac{1}{\varepsilon}$ or $\frac{1}{\varepsilon^2}$ and to the size of nodes served by each aggregator. Thus, the smaller is the tolerated error bound ε, the higher is the transmission overhead on the aggregator during the interactive verification phase. Performing the interactive verification directly with the network operator (one-hop communication), seems to be impractical and highly energy consuming especially for those aggregators that are far from it. Even using multi-hop communications remains still costly, because nodes in the path between a remote CH and the network operator will also forward the $48 \times \beta$ bytes.

Our protocol has an acceptable and less transmission overhead, comparing to the two other protocols. First, each sensor node in the cluster locally transmits (using one-hop communication) two packets, of 16-byte payload for each. Each CH transmits also two packets: one broadcast packet of 16-byte payload in the cluster using one-hop communication only, and one packet containing the aggregation result of $16 + \frac{1}{8}\log_2 k$ to the BS using multi-hop communication, and forward other aggregation results $(16+\frac{1}{8}\log_2 k)$ of some other clusters. As consequence, CHs in our protocol have less transmission overhead than in Przydatek et al. Protocol. Comparing to Hu et al. protocol, we must note that in Hu et al. protocol around half sensors of the network are aggregator nodes, while in our protocol based in Sun et al protocol [19], only around $5-7\%$ of sensors are aggregator nodes. As a result, half nodes in Hu et al. protocol will forward at least 40 bytes during aggregation phase, and participate in relaying the $8 \times n$ bytes of the disclosed keys in the delayed authentication phase, while in our protocol few aggregator nodes exist, and each aggregator (CH) forwards its aggregation result and some few other aggregation results of some distant CHs.

8 Limitations of Our Protocol

Our protocol mainly suffers from its restriction to static networks where new incoming nodes are rare (clusters are formed once all nodes are deployed), and where nodes are static once deployed. In addition, aggregation in our protocol is only done inside each cluster. The aggregation result of each cluster is sent to the BS, instead of serving as an input to the aggregation process of the next cluster. This seems to be the only way to protect the aggregation process from compromised and misbehaving aggregators. If the aggregation result of a cluster is aggregated again by the next (upper) aggregator, which is in the path to the BS, the BS has no way to check if the aggregation result of a particular cluster is valid or not. Moreover, the upper aggregator node has no way to check if the aggregation result sent by the down aggregator is valid or not.

9 Conclusion and Perspectives

This paper proposes a new secure aggregation protocol for WSN which does not raise on the usual restrictive condition that the aggregator nodes are trusted nodes, and which introduces little transmission overhead in the network, especially for CHs. Our protocol is resilient to the compromising of all aggregator nodes and part of nodes in the network. Our protocol distributes the task of aggregation inside a cluster, such that an attacker has no way to falsify the result of aggregation other than compromising the aggregator node and a significant portion of nodes in the cluster.

As a future work, our protocol will be implemented and its performances evaluated through .

Acknowledgement

Authors are thankful to French ANR (Agence Nationale de la Recherche) which funded RNRT project CAPTEUR.

References

1. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y.: Wireless sensor networks: a survey. Computer Networks (38), 393–422 (2002)
2. Karlof, C., Sastry, N., Wagner, D.: TinySec: A Link Layer Security Architecture for Wireless Sensor Networks. In: SenSys 2004 (November 3-5, 2004)
3. Perrig, A., Szewczyk, R., Wen, V., Cullar, D., Tygar, J.D.: Spins: Security protocols for sensor networks. In: Proc. of the 7th Annual ACM/IEEE International Conference on Mobile Computing and Networking, pp. 189–199 (2001)
4. Akkaya, K., Younis, M.: A survey on routing protocols for wireless sensor networks. Ad Hoc Networks 3, 325–349 (2005)
5. Krishnamachari, B., Estrin, D., Wicker, S.: The Impact of Data Aggregation in Wireless Sensor Network. In: Proceedings of the 22nd International Conference on Distributed Computing Systems, pp. 575–578 (July 2-5, 2002)
6. IntanagonwiI., C., Akyildiz, F., Su, W., Sankarasubramaniam, Y.: Wireless sensor networks: a survey. Computer Networks (38), 393–422 (2002)
7. Estrin, D., Govindin, R.: Impact of Network Density on Data Aggregation in Wireless Sensor. In: Proceedings of the 22 nd International Conference on Distributed Computing Systems, pp. 457–458 (July 2-5, 2002)
8. AI-Karaki, J.N., UI-Mustafa, R., Kamal, A.E.: Data Aggregation in Wireless Sensor Networks - Exact and Approximate Algorithms. In: Workshop on High Performance Switching and Routing, pp. 241–245 (April 19-21, 2004)
9. Hu, L., Evans, D.: Secure Aggregation for Wireless Networks. In: Proceedings of the 2003 Symposium on Applications and the Internet Workshops, p. 384 (2003)
10. Przydatek, B., Song, D., Perrig, A.: SIA: Secure Information Aggregation in Sensor Networks. In: SenSys 2003 (November 5-7, 2003)
11. Zhu, S., Setia, S., Jajodia, S., Ning, P.: An Interleaved Hop-by-Hop Authentication Scheme for Filtering of Injected False Data in Sensor Networks. In: Proceedings of the 2004 IEEE Symposium on Security and Privacy, pp. 259–271 (May 9-12, 2004)
12. Blundo, C., Santis, A.D., Herzberg, A., Kutten, S., Vaccaro, U., Yung, M.: Perfectly secure key distribution for dynamic conferences. In: Brickell, E.F. (ed.) CRYPTO 1992. LNCS, vol. 740, pp. 471–486. Springer, Heidelberg (1993)
13. Blom, R.: An Optimal Class of Symmetric Key Generation. In: Beth, T., Cot, N., Ingemarsson, I. (eds.) EUROCRYPT 1984. LNCS, vol. 209, pp. 335–338. Springer, Heidelberg (1985)
14. Chan, H., Perrig, A., Song, D.: Random Key Predistibution Schemes for Sensor Networks. In: IEEE Symposium on Security and Privacy, Okland, California, USA, pp. 197–213 (2003)
15. Dimitriou, T.T., Krontiris, I.: A Localized, Distributed Protocol for Secure Information Exchange in Sensor Networks. In: Proc. of the 19th IEEE International Parallel and Distributed Processing Symposium (2005)

16. Heinzelman, H., Chandrakasan, A., Balakrishnan, H.: Energy-efficient communication protocol for wireless microsensor networks. In: Proceedings of the 33rd Annual Hawaii International Conference on System Sciences (January 4-7, 2000)
17. Manjeshwar, A., Agrawal, D.: TEEN: A protocol for enhanced efficiency in WSN. In: Proceedings of the 15th International Parallel & Distributed Processing Symposium, pp. 2009–2015 (April 23-27, 2001)
18. Manjeshwar, A., Agrawal, D.: APTEEN: A hybrid protocol for efficient routng and a comprehensive information retrieval in WSN. In: Proceedings of the International Parallel and Distributed Processing Symposium, pp. 195–202 (April 15-19, 2002)
19. Sun, K., Peng, P., Ning, P., Wang, C.: Secure distributed cluster formation in wireless sensor networks. In: 22nd Annual Computer Security Applications Conference (December 11-15, 2006)
20. Grey, M., Jonhnson, D.: Computers and intracrbility: A guide to theory of NP-Completeness. W.H Freeman and Company, New York (1979)

Misbehaviors Detection to Ensure Availability in OLSR

Frédéric Cuppens, Nora Cuppens-Boulahia, Tony Ramard, and Julien Thomas

GET/ENST Bretagne, 2 rue de la Châtaigneraie, 35512 Cesson Sévigné Cedex,
France

Abstract. In this paper, we investigate the use of Aspect-Oriented Programming (AOP) [13] in the domain of Mobile Ad-hoc NETworks (MANETs). More precisely we study the availability issues in Proactive routing protocols. This paper classifies the different possible attacks and examines the countermeasures to ensure availability. Our approach is based on a detection-reaction process. The reasoning followed by the detection process is built on a formal description of normal and incorrect node behaviors. This model allows us to derive security properties. These properties are woven into our implementation using the AOP. Our algorithm checks if these security properties are violated. If they are, detection of incorrect (malicious) behaviors occurs to allow the normal node to find a path without incorrect node behavior. Therefore the detector node sends to its neighborhood the detection information to allow its neighbors to avoid choosing the intruder as a node to cross to. A node chooses the path using its local diagnosis and the reputation of other nodes. Using a field in the standard control message to communicate the detections, our approach does not change the message format, so it is very easy to use and there is no overhead. While we use OLSR as an example of protocol for our studies, we argue that the presented techniques apply equally to any proactive routing protocol for MANETs.

Keywords: Mobile Ad Hoc Network, Intrusion Detection, Availability, OLSR, Routing.

1 Introduction

A Mobile Ad-hoc NETwork (MANET) is a collection of nodes which are able to connect to a wireless medium forming an arbitrary and dynamic network. The routing protocol ensures that all nodes at all times can reach all destinations in the network. However several attacks can occur against security in order to disrupt the network.

In this paper, we investigate the issues of intrusion detection and response in MANET. As a main result, we provide a security extension to OLSR, a proactive MANET routing protocol. Our primary issue with respect to securing MANET routing protocols is to ensure the network integrity, even in presence of malicious nodes. It is not our propose in this paper to deal with node authentication which is an issue already investigated elsewhere[14]. Our approach is based on

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 799–813, 2007.

a formal security model called Nomad [7]. This model allows us to express node behaviors (normal and incorrect behaviors). From these expressions, we can derive properties to specify a security policy. These properties are woven into the routing protocol using an Aspect-Oriented Programming (AOP). These properties are checked when a message is received in order to detect intrusions. If a property is violated, a reaction occurs and the node attempts to find another path or Multipoint Relay (MPR) keeping the malicious node away. In this case, the node sends relevant information related to the detection to its neighborhood. The neighbors of this node record this information but do not fully trust it. A function allows nodes to compute the reputation in their neighbors. The reputation quantification allows nodes to choose the best path to reach another node.

The remainder of this paper is organized as follows. Section 2 presents the different kinds of Ad hoc routing protocol especially OLSR. Section 3 describes the vulnerabilities of Ad hoc routing protocols including OLSR. In section 4 we present related works. Section 5 gives an outline of our approach to satisfy availability requirements in ad hoc networks and briefly presents the modeling language we choose to express these availability properties and to specify node profiles. In Section 6, we define the node profiles and availability properties to detect and to communicate malicious behaviors and we show how these properties are woven with AOP into the code to secure the OLSR protocol. Section 7 is an experimentation of our mechanism to secure OLSR based on these properties and section 8 concludes.

2 Mobile Ad Hoc Network (MANET)

In Ad hoc networks, to ensure the delivery of a packet to a destination node, each node must run a routing protocol and maintain its routing tables in memory. Routing protocols can be classified into the following categories: reactive, proactive, and hybrid.

In this section, we present the Optimized Link State Routing protocol (OLSR) [4] using as an example to illustrate our approach. OLSR is a proactive routing protocol, designed specifically for large and dense MANETs. It is based on a Multipoint Relaying (MPR) flooding technique to reduce the number of topology broadcast packets, see figure 1.

2.1 Overview

Every node broadcasts HELLO messages that contain one-hop neighbor information periodically. If the Time To Live (TTL) of HELLO message is 1, the message is not forwarded. With the aid of HELLO messages, every node obtains local topology information.

A node (also called selector) chooses a subset of its neighbors to act as its Multipoint Relaying (MPR) nodes. This choice is based on the local topology information carried by HELLO messages. MPR nodes have two roles:

- When the selector sends or forwards a broadcast packet, only its MPR nodes among all its neighbors forward the packet;
- The MPR nodes periodically broadcast its selector list throughout the MANET in TC (Topology Control) message. Thus every node in the network knows by which MPR node the target node could be reached.

Notice that there is no guarantee that the selected MPR node is not a malicious node.

With global topology information stored and updated at every node, a shortest path from one node to every other node could be computed with Dijkstra's algorithm [8], which goes along a series of MPR node.

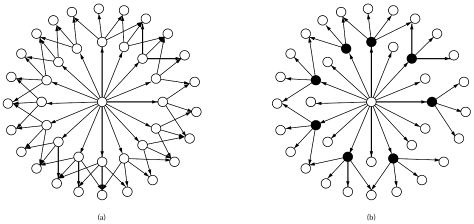


Fig. 1. Two hop neighbors and 'multipoint relays' (the solid circles) of a node. (a) illustrates the situation where all neighbors retransmit a broadcast, (b) illustrates where only the MPRs of a node retransmit the broadcast [4].

3 Security Flaws

In this section, we discuss various security vulnerabilities in ad hoc network.

One vulnerability, common to all routing protocols operating a wireless ad-hoc network, is 'jamming', i.e. a node generates massive amounts of interfering radio transmissions. In this paper, we do not consider a network resistance against jamming nor traffic overloading.

Attacks against MANETs can be divided into two groups: Passive attacks typically involve only eavesdropping of data whereas active attacks involve actions performed by adversaries, for instance the replication, modification and deletion of exchanged data. External attacks are typically active attacks that are targeted to prevent services from working properly or shut them down completely.

In summary, a malicious node can disrupt the routing mechanism employed by several routing protocols in the following ways. It attacks the route discovery process by generating link spoofing or identity spoofing, changing the contents of a discovered route, modifying a route reply message, causing the packet to be dropped as an invalid packet, invalidating the route cache in other nodes by advertising incorrect paths, refusing to participate in the route discovery process. The malicious node attacks the routing mechanism by modifying the contents of a data packet or the route via which that data packet is supposed to travel, behaving normally during the route discovery process but dropping data packets causing a loss in throughput.

These vulnerabilities makes it clear that ad hoc networks are inherently insecure, more so than their wireline counterparts, and need a mechanism to counter attacks on a system as soon as possible (ideally in real time) and take appropriate action.

4 Related Works

Sergio Marti et al. discussed two techniques that improve throughput in MANETs in the presence of compromised nodes that agree to forward packets but fail to do so [15]. A node may misbehave because it is overloaded, selfish, or broken. However, a node can only detect these behaviors and does not communicate this detection to its neighborhood. We take into account this approach, and we add the way to communicate the detection information to the neighborhood.

Bhargava et al. [3] proposed an intrusion detection and response model (IDRM) to enhance security in the Ad Hoc On Demand Distance Vector (AODV) routing protocol. When the misbehavior count for a node exceeds a predefined threshold, the information is sent out to other nodes as part of global response. However in this approach, a collusion of nodes can eject a normal node from the network. In our approach we adopt the communication of the detection information. A node sends periodically to its neighborhood the trust level of each neighbor.

In [2] the authors propose to associate one signature with each OLSR message, rather than one for each OLSR packet. In addition to this, one timestamp is provided for each signature. The timestamps are used to assess the freshness of the messages, thus avoiding replay attacks. The signature is encapsulated and transmitted as an ordinary OLSR message. This means that the signature and the message can travel in separate packets and separate routes from the originator. Also, the proposed system in [2] is an end-to-end system. The suggested timestamp exchange protocol proposed in [2] is a rather complex solution.

Most of the research works (like [12] [11]), attempt to apply cryptography techniques to secure MANET routing protocols. But, we know, as in wired network, that in addition to intentional and not intentional malicious behaviors there are always design flaws, human errors that enable attackers to exploit software vulnerability. Hence, we follow a property oriented intrusion detection approach and develop a reaction mechanism to deal with the detected intrusions. The difference between our approach and these related works is the fact that we do not change the message format. Consequently our algorithm is easier to implement. In section 5 we explain the model we use to define the node profiles and the properties woven in our AOP approach. These properties are the orthogonal aspect in our approach.

5 Modeling Approaches

To study different availability properties in mobile ad-hoc networks (MANET), we take into account topological dimension. This study is based on the properties of topological information exchanged during the network building and the topological maintenance. The regular network maintenance between nodes allows the discovery of the available routes and the participant nodes. Each node provided with a sensor, analyzes and detects errors in "control messages" exchanged between the nodes and readjust, if possible, its routing tables in compliance with this analysis.

When dealing with security properties like availability, classical first order logics are no longer appropriate. We need a more expressive security model such as the Nomad model [7]. Nomad is a security model based on deontic logic and temporal and temporized logics of actions which provides expressiveness necessary to specify availability requirements. Thanks to the Nomad model, we specified the OLSR protocol and expressed availability properties. In this paper, we only use the temporal and temporized framework of Nomad. Thus, we introduce the temporal modality $\Box A$ and the temporized modality $\bigcirc^{\leq d} A$, for $d \geq 0$. If A is a formula, then $\Box A$ is to be read "A is always true" and $\bigcirc^{\leq d} A$ is to be read "A is eventually true within a delay of d units of time". Using these modalities, we can express two availability properties:

- "Usual" availability: a message m must be received by a node nd in maximum delay d each time it is sent by a node ns

$$(a) \quad \Box(SEND(ns, nd, m) \rightarrow \bigcirc^{\leq d} RECEIVE(ns, nd, m))$$

- Weak availability: a message m must be received by a node nd in maximum delay d each time it is sent by a node ns and there exists a route, a transitive closure of symmetric links, between ns and nd.

$$(b) \quad \Box(SEND(ns, nd, m) \wedge ROUTE(ns, nd) \\ \rightarrow \bigcirc^{\leq d} RECEIVE(ns, nd, m))$$

We try to satisfy the property (b), as it is the most compliant availability property with the characteristics of ad-hoc networks. For this purpose, we shall derive more basic properties (see the following section) from the protocol specification.

6 OLSR Availability Analysis

Each node has a view of the network topology derived from (Hello and TC) messages it receives. This view can be modified by a malicious node and an attack against the availability can occur. To study different availability properties, node profiles have to be specified (6.1) to understand the behavior of normal and malicious nodes. Thanks to theses profiles and the messages, basic properties (6.2) can be expressed and have to be satisfied to ensure availability.

6.1 Node Profiles

MANET nodes which participate in the network routing can be grouped by the way they act in the network. The identified behaviors of nodes are called node profiles. These profiles are very important to understand how a normal node and an intruder work. Thus, we can derive properties to identify theses profiles.

- Profile of a cooperative MPR node nb who always transmits TC messages when it receives them from its neighbor na before the expiration of time Max.

$$\begin{aligned}\boldsymbol{COOPERATIVE(nb)} \leftrightarrow \Box(RECEIVE(na, nb, m)\wedge \\ NEIGHBOR(nb, nv) \wedge MPR\_NEIGHBOR(na, nb) \\ \rightarrow \bigcirc^{\leq Max} PROPAGATE(nb, nv, m))\end{aligned} \tag{1}$$

- Profile of a “lazy node” nb who transmits messages irregularly.

$$\boldsymbol{LAZY(nb)} \leftrightarrow \neg COOPERATIVE(nb) \tag{2}$$

- Profile of an “selfish node” nb who never transmits messages. An egoist MPR node receives TC messages directly from the sender or other MPR relays but it does not transmit those messages.

$$\begin{aligned}\boldsymbol{SELFISH(nb)} \leftrightarrow \Box(RECEIVE(na, nb, m)\wedge \\ MPR\_NEIGHBOR(na, nb) \wedge NEIGHBOR(nb, nv)) \\ \rightarrow \Box\neg PROPAGATE(nb, nv, m)\end{aligned} \tag{3}$$

- Profile of a “slanderer node” that generates incorrect information. Such a node can forward incorrect information (carried by control messages) received from other nodes.

$$\begin{aligned}&\boldsymbol{SLANDERER(nb)} \leftrightarrow \\ &(\neg TC(na, nb, m) \wedge NEIGHBOR(nb, nv)\wedge \\ &MPR\_NEIGHBOR(na, nb) \wedge PROPAGATE(nb, nv, m)) \\ &\vee \\ &(\neg HELLO(na, nb, m) \wedge NEIGHBOR(na, nb)\wedge \\ &SEND(nb, na, m))\end{aligned}$$

- Profile of a “secretive node” (malicious MPR node) that does not forward any correct message which has to be forwarded through this node.

$$\begin{aligned}&\boldsymbol{SECRETIVE(nb)} \leftrightarrow \\ &(TC(na, nb, m) \wedge BELIEVE(nb, m)\wedge \\ &MPR\_NEIGHBOR(na, nb) \wedge NEIGHBOR(nb, nv)\wedge \\ &\neg PROPAGATE(nb, nv, m)) \\ &\vee \\ &(HELLO(na, nb, m) \wedge BELIEVE(nb, m)\wedge \\ &NEIGHBOR(na, nb) \wedge \neg SEND(nb, na, m))\end{aligned}$$

- Profile of a "liar node" nb that generates incorrect information or does not forward any correct message which has to be forwarded:

$$\boldsymbol{LIAR(nb)} \leftrightarrow SLANDERER(nb) \vee SECRETIVE(nb) \tag{4}$$

- Profile of a "honest node" nb that sends only correct routing information:

$$\boldsymbol{HONEST(nb)} \leftrightarrow \Box\neg LIAR(nb) \tag{5}$$

Among these profiles we are particularly interested in the profile of liar node that we use to derive the properties shown in the section 6.2.

6.2 Basic Properties Specification For Detection Of Liar Nodes

Using the characteristics of OLSR (section 2), we can derive some properties that allow us to detect the inconsistencies in OLSR control messages.

Hello-TC Relationship Property: For a MPR node, all its MPR selectors carried by TC messages are always found among all the one-hop neighbors carried by Hello message.

$$\begin{aligned} HELLO(na, nb, m) \wedge TC(na, nb, m') \wedge NEXT(m, m') \\ \wedge MPR(m', n_c) \rightarrow NEIGHBOR(m, n_c) \end{aligned} \tag{6}$$

MPR-MPR Selector Relationship Property

- When a node nb receives a TC message from node na, if node nb is claimed as node na's MPR selector, then node nb must have chosen node na as its MPR.

$$\begin{aligned} TC(na, nb, m) \wedge IN\_MPRS\_SET(nb, m) \\ \rightarrow MPR\_NEIGHBOR(nb, na) \end{aligned} \tag{7}$$

- When a node nb receives a TC message from node na, if another node nc is claimed as MPR selector of node na, then node nc must have chosen node na as its MPR and declared that in its previous Hello message.

$$\begin{aligned} TC(na, nb, m) \wedge HELLO(nc, nb, m')\wedge \\ IN\_MPRS\_SET(nc, m) \rightarrow IN\_MPR\_SET(na, m') \end{aligned} \tag{8}$$

Message Integrity Property: When a MPR node $n1$ receives a TC message and if this message must be forwarded via node $n1$, then the TC message must not be modified by node $n1$. The same copies of the TC message must be received by its originator and all MPR nodes who have forwarded this TC message.

$$\begin{aligned} TC(ns, n1, m) \wedge MPR\_NEIGHBOR(ns, n1) \\ \rightarrow TC\_RELAY(n1, m) \end{aligned} \tag{9}$$

For instances the message integrity property allows a node N to ensure the integrity of TC messages exchanged in the MANET. Node N sends, in its TC message, a list of nodes $MPRS\_SET = \{A, B, C, D\}$ that have selected node N as their MPR with the sequence number equal to a value p. The TC message is forwarded by node D and then by node E in order to reach the whole MANET. This case could present two types of possible attacks on the payload of the TC message:

- Modification of the list of MPR selectors: if node D (respectively E) is a lying node and tries to modify the content of the TC message sent by node N, the node N (respectively D) will detect this intrusion.
- Modification on the sequence number: the intruder node D (or E) forwards the received TC message by modifying the sequence number into another value $p'(p' >> p)$. Consequently, nodes E and F stop treating any TC messages originated from A with a value lower than p'.

When a node receives a Hello or TC message, it applies these properties to check the validity of the received message. If the property is violated, then many attacks are possible: (1) The message sender has lied and wished to be selected as a MPR (Link spoofing), (2) The message sender has lied on its identity (identity spoofing) or (3) some relays or an intruder along the way between the source of TC message and the receiving node could also modify the message.

The properties (defined in Basic properties specification for detection of liar nodes) can only detect a "liar node" described in section 6.1. Unfortunately, our detection process based on these properties works well in the case of "information redundancy". So we investigate other properties to detect selfish profiles, and a way to allow a node to send the detection information to its neighborhood. Now we introduce how we can detect these profiles even if there is no information redundancy.

Interval Transmission Property: When a node A selects node B as MPR. the node B must send a TC message with A inside. The emission interval of TC is defined in the TC message (by default this interval is 5 seconds). So, if the node B does not send this TC message before this delay, the node A can detect a "lazy node" or a "selfish node". All common neighbors of A and B can also detect this profile, because they also received the Hello message from A, and they can check if B sends a TC message. An example is shown in figure 2(a).

$$\begin{aligned} &HELLO(na, nb, m) \wedge IN\_MPRS\_SET(nb, m) \\ &\quad \rightarrow \bigcirc^{\leq TC\_INTERVAL} TC(nb, nv, m') \end{aligned} \tag{10}$$

In the same way, a node can detect if a TC message is forwarded or not before a $TC\_INTERVAL$. The TC message is broadcast in the whole network by the MPR node. So when a MPR node forwards a TC message, this node checks if this message is forwarded before some delay. If this message is not forwarded the node detects the "selfish node". However the presence of ambiguous collisions, receiver collisions, limited transmission power, collusion, and partial dropping are detected as ”selfish node” whereas they are not. But if there is a collusion, or a limited transmission power, or partial dropping. Therefore the source is not very sure that the packet arrives to the destination. Thus, it is better to change the intermediate nodes to reach the destination.

$$\begin{aligned} & TC(na, nb, m) \wedge MPR\_NEIGHBOR(na, nb) \\ & \rightarrow \bigcirc^{\leq TC\_INTERVAL} PROPAGATE(nb, nv, m) \end{aligned} \tag{11}$$

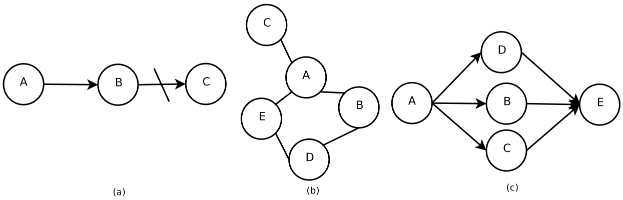


Fig. 2. (a) An example of a ”selfish node”. (b) No node can check if the node C is really a neighbor of B . (c) To reach the node E, the node A chooses the node which has the greater willingness.

Neighbor Relationship Property: When the local node A has a 2-hop node C reachable by only one neighbor B, it means that B is a MPR of A. But there is no way to know if this link between B and C exists. To avoid the impact of B, the node A decreases the confidence in node B. Thus, the node A chooses other MPR whose confidence is higher. Notice that the node B is still a MPR of A. An example is shown in figure 2(b).

Willingness Property: The willingness field is a parameter exchanged in the Hello message [4], thus only the 1-hop neighbors receive this parameter. The willingness field has a value by default, so when a new node arrives in the network it has the default value therefore the new node is not isolated, its (control) messages are exchanged or forwarded. However, when a node finds an intrusion, the node informs about this detection according to the willingness field. So when the willingness is low, it means that the node has incorrect or malicious behavior. When a node A receives an information from the node B, where the node C is claimed as a malicious node (because its willingness is low). the node A applies a reputation function (see below) to deal with this information. Our function is based on [5] that we modify to be in compliance with our topic. The willingness is computed every time the node receives a Hello or TC message.

$$w_x = w_1 + w_2 \tag{12}$$

Where w_1 is the checks over the properties defined in this section and w_2 is the detection information from its neighbors.

$$w_1 = w_{Lx} * p \tag{13}$$

$$w_2 = 1/N * (\sum_{k=0}^{N}(w_{Lk} * w_{kx})/w) * (1-p) \tag{14}$$

With:

- w_x is the final willingness in the neighbor x. This willingness is used to choose the MPR nodes.
- w_{Lx} is the willingness of the local node in the neighbor x. If these properties (defined in 6.2) are violated by the node x, then the willingness changes. We only choose to take into account the most recent information about the nodes. Thus, every time the node receives a message, this value decreases if a property is violated. w_{Lx} increases if no property is violated.
- w_{Lk} is the willingness of the node in the neighbor k. The greater w_{Lk} is, the greater w_{kx} impact is.
- w_{kx} is the willingness of the node k in the neighbor x
- w is the willingness by default defined in OLSR specification.
- N is the number of neighbors
- p and $(1-p)$ is the weighting. Here, $p \geq 0,5$, and $(1-p) = Min\langle(\lfloor N/3 \rfloor); 0,5\rangle$. Where $(1-p)$ is the minimum between the number of neighbors divided by three and $0,5$. If a node has less than three neighbors, we do not take into account the information from the neighborhood, because we do not have enough neighbors to make a good average of the willingness. Thus, more the node has neighbors, more they have influence.

In the RFC of OLSR [4], the 2-hop neighbors is the only parameter to select the MPR node. With our approach the willingness and the 2-hop neighbors help the choice. Hence, the choice is better and allows the local node to keep away the intruders. In Figure 2(c), the node A must choose one of B, C, D to be its MPR. For that, A chooses the node which has the greater willingness. If the node A does not detect any intrusion, the MPR is randomly selected. To avoid this issue, the node takes into account the detection information from its neighbors.

6.3 Our Algorithm For Profile Identification

Our mechanism can be implemented on each OLSR node in order to detect conflicts or inconsistencies in the OLSR control messages. When a node receives a TC message, it uses these properties to check and validate the TC messages before it updates its routing tables. A node which sends TC message and uses these properties to detect if there is an anomaly during the exchange of routing information. By doing so, the security level and the robustness of routing operation can be optimized. To allow normal node to choose another path without intruders.

For this purpose, we weave the properties defined in section 6.2 in the code using an AOP approach. As each property is checked when a message is received, a property is used to identify some pointcuts. We weave at these pointcuts our detection and reaction mechanism in the same manner as we did in wired networks [6] for securing the TCP/IP protocol.

In the weaving approach (see figure **??**), the functional aspect is the OLSR protocol, especially the message reception specification part.The information about the control message is used to update the Topology table, and then the MPR nodes and routes are chosen.

As long as there is no detection, the OLSR algorithm does not change (see figure 3). When a willingness is updated or a property is violated, the algorithm for profile identification is applied. First, our algorithm checks the properties defined in section 6.2. The algorithm then identifies the node profile and computes the willingness according to this profile. The last aspect in figure 3 changes the list of nodes using computation of willingness and then to see to it that only non malicious nodes are chosen. Sending TC or Hello message with the new willingness, the neighbors can take this information into account.

Our mechanism checks several properties to find a reliable path to the destination. However the complexity of this algorithm is in $O(n^2)$ where n is the number of 1-hop symmetric neighbors, but we optimized this complexity using the hashmap to get a complexity in $O(1)$. Therefore, when a node A receives a control message from node B, it checks if the 1-hop symmetric nodes identify B as an 1-hop symmetric node. Therefore the time cost is $n * O(1) = O(n)$. Moreover, the maximum interval transmission between two control messages is the interval transmission between two TC messages, by default this interval is 5 seconds [4]. Therefore a detection is made and communicated in less than 5 seconds.

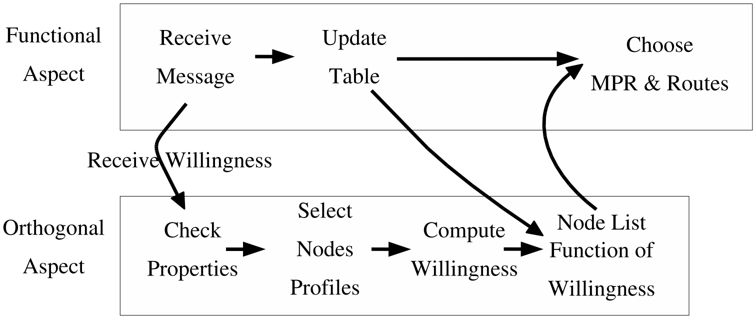


Fig. 3. Weaving functional and orthogonal aspects

7 Availability Experimentation in OLSR

We simulate a Mobile Ad hoc NETwork (MANET) in section 7.1 to test our algorithm. In section 7.2, we illustrate how the algorithm works, and which attacks it can detect. Also, we show that a node can reach another node even in the presence of a malicious node. In section 7.3, we deal with some extreme cases, where a malicious node succeeds in preventing some node from reaching another node.

7.1 Experimental Virtual Network

To simulate a network in our experimentation, we use the User-Mode Linux [9] to create nodes and Iptables [10] to make the links between nodes.

User-Mode Linux provides a virtual machine. User-Mode Linux is a safe, secure way of running Linux versions and Linux processes.

After creating the nodes, we simulate the physical link between them. For this purpose, we use Iptables to write rules that accept packets from the neighbor nodes and block other traffic.

Thanks to User-Mode Linux and Iptables we obtain the network presented in figure 2(c). We choose this topology to test the impact of an intruder like "Liar node", "Selfish node". We assume that the links between are not noisy and the nodes are not very mobile. If there are noisy, this problem is dealt with by the lower layer or the routing protocol itself. If the nodes are mobile therefore the nodes generates incorrect information and it would be selected as a liar node in a first time. After a delay it would not generate other incorrect information and its willingness would be greater and it would be selected as a normal node by its neighbors. The main of our approach is to select the good neighbor to forward the message.

7.2 Analysis

We present some results of our simulations using the example of figure 2(c). The simulation results show the contents of routing tables for each node of the chosen topology: (1) activated analysis mechanism and (2) deactivated analysis mechanism. We then analyze the topology with the normal node behavior, and finally the topology with an intruder and without the analysis mechanism, and the topology with an intruder and the analysis mechanism.

Normal Node Behavior Simulation: We started our simulation with the normal behaviors of nodes without any attack and any verification, and figure 2(c) summarizes the routes used in the network. In this case, we supposed that the quality of all the network links was perfect (without packet loss). The choice between 2 neighbors which have the same 2-hop neighbors is random, because there is no other parameter to help the choice. Table 1 presents relevant OLSR data obtained for the example.

Attacks Simulation: In figure 4, in the first case, the nodes A, B and E chose only the node D as MPR because D has created a fictive link with a known or unknown node. In the second attack, the node D does not forward the message from A to E. The node A and common neighbors of A and D do not detect this attack and the node D stays a MPR node. In the third case, the nodes D and B make a wormhole attack, and they claim that the node E is their neighbor. In this case, the node D stays the MPR of A, and the node C chooses the node B as MPR. So the node B and D become MPR. Thank to this attack, the node B and D obtain privilege.

Table 1. Relevant OLSR data

Node	1-hop neighbors	2-hop neighbors	MPR	MPR selectors
A	B,C,D	E	D	C
B	A,B,C,E	-	-	D
C	A,B,E	D	A	-
D	A,B,E	C	B	A,E
E	B,C,D	A	D	-

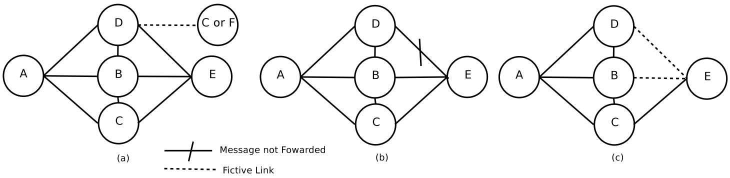


Fig. 4. In (a) the node D creates a link with a known or unknown node, in (b) the node D is a selfish node, in (c) The node D and B make a wormhole attack

Use Of The Analysis Mechanism: In this step, all nodes (except the intruder) run the same OLSR protocol in which our detection and response mechanism is implemented. In figure 4, in the first case the nodes A, B and E detects that the node D has a link with an unknown node (Neighbor relationship property defined in 6.2). So they decrease the willingness in the node D and thus choose another MPR. So The node A and E chose B or C as MPR. But the node D stays the MPR of A, B and E because the node F is its neighbor. This limits the impact of the attack because the nodes chose another node as MPR to reach their 2-hop neighbors.

In the second attack, the node D does not forward the message from A to E. In this case the node A and B detect that the node D is a "selfish node" (Property of transmission intervals defined in section 6.2). Therefore the node A chooses the node B or C as MPR. Decreasing the willingness of D, the node B sends this detection information to the node E and the node E computes the new willingness (Willingness property defined in section 6.2) and chooses B or C as MPR. We obtain the same result when "liar node" occurs, for example if the node D claims that C is in its neighborhood. This example shows that our approach provides means to choose the good MPR despite the presence of "selfish node" or "liar node".

In the third case, the node C detects that the node B is a "liar node" (Hello-TC relationship or/and MPR-MPR selector relationship property defined in section 6.2), and sends this detection information to the node A. The node A decreases the willingness of the node B (Willingness property defined in section 6.2). So the node A chooses between C and D to be its MPR. At time t_0, the choice is random but at time t_1 the node A chooses the node C. The node C

sends a TC message from E to A and the willingness of A in C increases and is greater than B. If the node A has more neighbors, this makes easier the choice of a good MPR, so at time t_0 the choice is only C or another good node. This shows that, using our approach, the wormhole attack does not prevent a non malicious node from reaching another node.

If a route exists between two nodes, then the nodes are reachable even if an intruder tries to change or to block the route. We plan to test in our future works several mobility models using the network simulator ns2 [1].

7.3 Extreme Cases

In this section, we show that there are cases impossible to solve. For instance, if the node has only intruders as neighbors, there is no solution.

Another extreme case looks like the third case in the figure 4, where a wormhole occurs. But here, the willingness of B in C is close to 0. The willingness of B in D and the willingness of B in D are maximal and the node B or D simulates a fictive TC message from E. So the node A chooses D as MPR. However, the problem disappears when there is a larger number of non malicious nodes because the node will take other willingness into account and will choose C or another good node.

8 Conclusion

The techniques presented in this paper are based on specifying security properties in MANET, especially the availability property. If a route exists from a mobile node to another, then this node (if it is permitted) would be able to obtain the route whenever it needs. And the routing operation would take a bounded delay to complete.

Through this study, we chose the OLSR protocol to analyze the availability requirements for MANETs. Several properties related to availability have been expressed based on the specification of the protocol OLSR (these properties are compliant with the RFC3626) and malicious node profiles are used to deploy an intrusion detection and reaction technique. Each MANET node observes its neighbors' behaviors corresponding to the received messages which provides means for checking if its neighbor is malicious or not. If a detection occurs, the node sends this information to its neighborhood. This approach seems to us the most adapted for MANETs. Aspect-Oriented Programming (AOP) makes easier the implemention of availability properties. AOP allows us to keep the standard OLSR specification unchanged when there is no detection and to use our algorithm when a detection occurs. The AOP approach allows us to define a method to secure any routing protocols provided we have specified security aspects to be woven in the protocol. To validate our analysis, an experimentation has been done on a virtual network where the analysis mechanisms and several misbehavior have been implemented. The obtained results from our experiments encouraged us to go further in our investigations. We plan to test our approach in a network simulator to take into account several mobility models. The objective is to express other properties that we shall use to adapt our detection/reaction mechanism.

Acknowledgement. This work was supported by the RED Celtic project and a French DGA (Direction Générale de l'armement) grant.

References

1. A collaboratoin between researchers at UC Berkeley, LBL, USC/ISI, and Xerox PARC. The ns Manual, http://www.isi.edu/nsnam/ns/doc/index.html
2. Adjih, C., Clausen, T., Jacquet, P., Laouiti, A., Muhlethaler, P., Raffo, D.: Securing the OLSR protocol. In: Med-Hoc-Net 2003. 2nd IFIP Annual Mediterranean Ad Hoc Networking Workshop, Mahdia, Tunisia (June 25–27, 2003)
3. Bhargava, S., Agrawal, D.P.: Security enhancements in aodv protocol for wireless ad hoc networks. In: VTC 2001 Fall, vol. 4, pp. 214–347 (2001)
4. Clausen, T., Jacquet, P. (eds.): Optimized Link State Routing protocol (OLSR) (2003) RFC 3626, http://www.olsr.org/
5. Conrad, M., French, T., Huang, W., Maple, C.: A lightweight model of trust propagation in a multi-client network environment: To what extent does experience matter? In: ARES, Vienna, Austria, IEEE Computer Society, Los Alamitos (2006)
6. Cuppens, F., Cuppens-Boulahia, N., Ramard, T.: Availability Enforcement by Obligations and Aspects Identification. In: ARES, Vienna, Austria (2006)
7. Cuppens, F., Cuppens-Boulahia, N., Sans, T.: Nomad: A Security Model with Non Atomic Actions and Deadlines. In: The computer security foundations workshop (CSFW), Aix en Provence, France (2005)
8. Dijkstra, E.W: A note of two problems in connection with graphs. Sumerisclie Mathematik 1, 269–271 (1959)
9. Bovet, D.P., Cesati, M.: Understanding the Linux Kernel. O'Reilly (2003)
10. Purdy, G.N.: Linux Iptables Pocket Reference. O'Reilly (2004)
11. Hong, F., Hong, L., Fu, C.: Secure OLSR. In: AINA 2005. 19th IEEE International Conference on Advanced Information Networking and Applications, Tamkang University, Taiwan (March 28-30, 2005)
12. Isida, S., Ando, E., Fukuzawa, Y.: Secure routing functions for OLSR protocol. In: 2005 OLSR Interop and Workshop, Palaiseau, France (July 28–29, 2005)
13. Kiczales, G.: Aspect-oriented programming. ACM Comput. Surv., 28(4es) (1996)
14. Lampson, B., Abadi, M., Burrows, M., Wobber, E.: Authentication in Distributed Systems: Theory and Practice. ACM Transactions on Computer Systems 10(4), 265–310 (1992)
15. Marti, S., Giuli, T.J., Lai, K., Baker, M.: Mitigating routing misbehavior in mobile ad hoc networks. Mobile Computing and Networking, 255–265 (2000)

Securing the Wireless LANs Against Internal Attacks

Ghassan Kbar and Wathiq Mansoor

American University in Dubai (AUD), UAE
{gkbar,wmansoor}@aud.edu

Abstract. Deploying wireless LANs (WLAN) at large scale is mainly affected by reliability, availability, performance, and security. These parameters will be a concern for most of managers who want to deploy WLANs. Most importantly, the security issue became the predominant factor in WLAN design. Different Intrusion detection mechanisms have been addressed in research papers, but with little being focused on internal intrusion and prevention. In this paper an efficient security method has been proposed. It is based on detecting rogue access points as well as rogue bridge access points and denying their access to the WLAN. In addition a new method of mutual authentication between DHCP server at the AP and wireless client has been introduced. This would allow client to detect rogue DHCP server and stop the association with it. It also allows registered DHCP server to detect unauthorized client and deny its request. Moreover the DHCP server would synchronize with the AP or intelligent LAN switch to drop packets from unauthorized client who might use static IP to get access to the network.

Keywords: wireless, security, intrusion detection & prevention, rogue AP, rogue DHCP.

1 Introduction

WLAN technology is rapidly becoming a crucial component of computer networks that widely used in the past few years. Wireless LAN technology evolved gradually during the 1990s, and the IEEE 802.11 standard was adopted in 1997 [1]. Companies and organizations are investing in wireless networks at a higher rate to take advantage of mobile, real-time access to information.

Enterprise managers want to deploy wireless networks with several important qualities. These include; high security, highly reliable and available WLANs with very little downtime, and high performance. The ideal wireless network is to have reliability, availability, security, and performance criteria to be similar of wired enterprise networks. In addition, it should be possible to deploy wireless networks very quickly and without the need for extensive and time-consuming site surveys. Furthermore, the networks should have the flexibility needed to support load balance and changes in the radio environment. Radio Resource Management (RRM) forms the basis of Quality of Service (QoS) provisioning for wireless networks [2]. It is an intense research area due to the wireless medium's inherent limitations and the increasing demand for better and cheaper services. Improving the mobility

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 814–821, 2007.

management has been addressed in [3] based on dividing the location management into two levels, intra and inter mobility. This will reduce the amount of signaling traffic, but still didn't address the problem of reliability and availability. Supporting security, reliability and QoS in dynamic environment has been discussed in [4] using modified routing protocol OSPF-MCDS over WLANs.

WLAN security is mainly affected by three internal attacks. These are rogue AP, rogue DHCP server, and unauthorized clients using static IP address. Firstly, a rogue AP allows unauthorized clients to access the corporate network and consume its bandwidth. Secondly, a rogue DHCP server through AP or stand alone server would cause problem to wireless clients in accessing the resources on the network safely. Where, a man in the middle attack can use the rogue DHCP server to route traffic to different destination. Thirdly, a wireless clients with static IP can still access the network and consume its bandwidth without been authenticated by the servers. Addressing the WLAN security to detect a rogue AP has been covered in [6], but it only covers AP wants to join the LAN switch. Intrusive actions and security challenges may originate outside *or within* a network [7]. This paper illustrates the use of snort to detect intrusion but using manual method to filter such attacks according to specified rules maintained by the administrator. Improving the performance of intrusion detection approach has been done at [8] but using a similar method of manual calibration to filter an attack based on signature with context. The necessity for intrusion detection techniques for mobile wireless network has also been covered in [9]. They proposed the use of anomaly detection models constructed using information available from the routing protocols for intrusion detection purposes. In this paper a new security mechanism has been suggested to cover Rogue Bridge AP that wishes to join existing corporate WLAN through existing AP without the use of LAN switch as described in section 2.1. Addressing the rogue DHCP has been covered by [10, 11] based on authenticated DHCP server. This requires both client and server supporting the authenticated DHCP server. Radek in [12] suggested forcing the usage rules in public WLANs to stop the effect of rogue DHCP. They used active and passive method to detect the rogue DHCP server based on list of official DHCP servers, or through authenticated DHCP server. They have not cover the situation where users get invalid IP address from the rogue DHCP since the monitoring done on the administration servers and not by the clients. A proper solution has been suggested in this paper which allows normal DHCP server to integrate with authentication server to provide mutual authentication between clients and DHCP server. This would allow the DHCP server to detect unauthorized clients and deny its request from getting a dynamic IP. In addition it allows client to detect a rogue DHCP server and decline its association with it to get a dynamic IP. Then if the client detects the rogue DHCP server, it would scan for a legal DHCP server who can be authenticated and trusted as described in section 2.2. Furthermore, a new mechanism has been suggested in this paper to synchronize the DHCP server with AP or intelligent switch to drop any packets coming from unauthenticated clients. This can be done at the AP by filtering packets that has IP address not registered in the DHCP list as valid IP. It can also be based on filtering packets that has registered IP address but with different MAC address not matching the one in the DHCP list. This will prevent wireless clients with static IP from utilizing the network bandwidth as explained in section 2.3.

2 Securing WLAN

Securing a network becomes nightmare due to the increase of attacks and the use of new techniques that are based mostly on free tools. This becomes even worse when it comes to WLAN that facing extra mean of attacks since its signal is broadcasted on a free media "the air". Attacks are not limited to external but also to internal which becomes hard to detect. Three of internal main attacks can cause a big problem to WLAN at the AP and the MTs. These are rogue AP, rogue DHCP server, and illegal use of bandwidth by unauthorized clients. In section 2.1, a new method has been suggested to improve the WLAN security by detecting and isolating a rogue APs. Section 2.2 describes a new method of integrating authentication with normal DHCP server to detect a rogue DHCP server by the client, and to deny unauthorized client from getting a dynamic IP if it fails the authentication. Section 2.3 describes a new mechanism of detecting unauthorized clients using static IP and drops their packets. This will save the network bandwidth from been utilized by illegal clients.

2.1 Securing the Network Against Rogue AP

Security at WLAN can be achieved using access control, Wired Equivalent Privacy (WEP), and authentication that can be based on dynamic WEP key, Kerberos authentication, Wi-Fi Protected Access (WPA). Despite using the above security mechanisms, security vulnerabilities still exist in WLAN based on IEEE 802.11. These vulnerabilities can be related to Denial of Service attack (DoS) and illegal use of AP that is called rogue wireless access point. A rogue access point is any Wi-Fi access point connected to a WLAN network without authorization. A rogue AP allows anyone with a Wi-Fi-equipped device to connect to a corporate network, leaving the IT assets wide open for the casual snooper or criminal hacker.

In order to address the security issue and protect against rogue access points, Proxim [6] implemented a way to monitor the wireless environment as well as the wired side of the network. They suggested integrated rogue AP detection in ORiNOCO AP where Rogue AP detection is accomplished through low-level 802.11 passive and active scanning functions for effective wireless detection of Access Points in its coverage area. In Proxim, security has been improved to detect the rogue AP, but however it didn't address a rogue bridge AP that will be joining an existing AP to extend the coverage distance.

In order to address the issue of Rogue Bridge Access Point based on the distributed management APs as proposed in [5], a new solution has been proposed in this section. This solution is based on validating the new joining AP Bridge according to their MAC address. Each AP belong to the same network would have a list of manageable APs that contain their MAC addresses. As shown in Fig 1, any time a rogue AP try to join the existing network through a manageable AP (eg. AP 4), its MAC address will be checked and accepted only the ones found in the list. In this scenario, the rogue bridge AP is controlled at the distributed manageable AP through rejection or acceptance during the join of new AP.

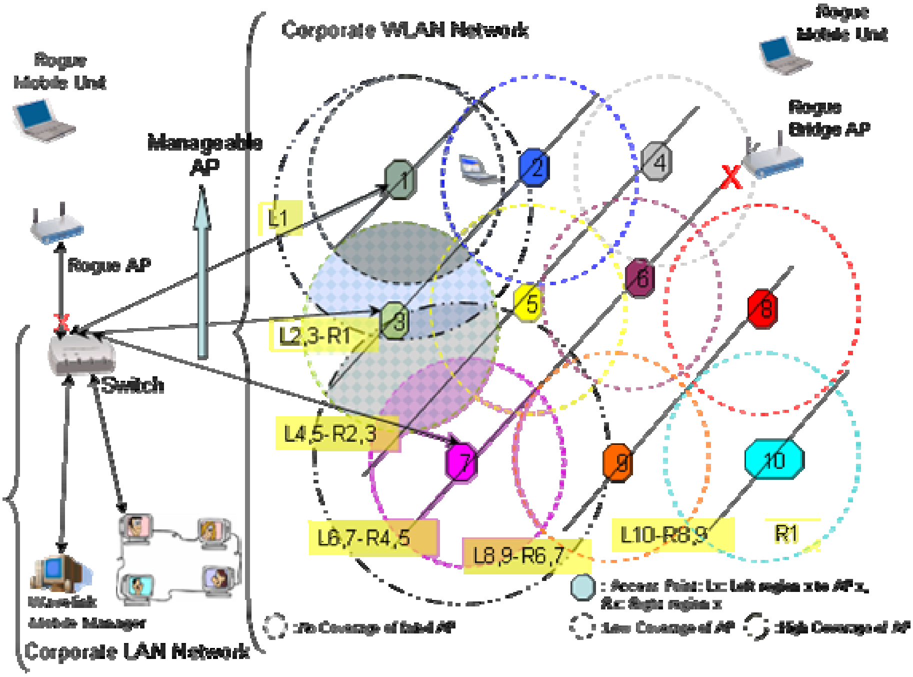


Fig. 1. Rogue AP attempts to join distributed management APs in a corporate WLAN

Once a new AP been accepted to join the distributed management APs through an existing AP (eg. AP 4), a dynamic WEP authentication key would be assigned to the new AP to become part of the corporate WLAN network. There will be master AP that would be configured at the initial stage that includes a list of all MAC addresses of possible AP might join the network. The list of manageable APs will be configured with this Master AP of the corporate WLAN at the initial stage and would be broadcasted to every other AP that is in the list and is part of the corporate network. If additional AP not included in the list and would be added to the corporate WLAN in the future, the master AP manageable list need to be updated manually by the administrator. Each AP joined the corporate WLAN would have two lists; a manageable list that contains the MAC addresses of all permissible AP, and current list that contains the MAC addresses of existing APs which joined this network and its location within the network map. The current list would be used to detect a failure of an AP, then determine its location and adjust the neighbor APs to increase the transmission power to allow high coverage area that substitute the failed AP. While the manageable list would be used by the existing AP to detect a rogue bridge AP and therefore stop it from joining the corporate WLAN. This will increase the network security and prevent and rogue mobile terminal from accessing the corporate WLAN illegally.

2.2 Securing the Network Against Rogue DHCP Server

Most of the current DHCP implementations don’t use authentication when servicing clients that want to obtain dynamic IP. This causes a security concern where unauthorized DHCP server (rogue DHCP server) might be connected to corporate network and pass false information to client. The false information might lead to different type of attacks including denial of service (DoS) attack and spoofing. It can also cause disruption of network usage where employee might not be able to use the corporate network because of passing wrong DNS information by the rogue DHCP server to clients. In addition, it allows unauthorized clients masquerading a valid client and access the network. This will cause loss of bandwidth and threat to WLAN. Internet Engineering Task Force [10] revised the new DHCP protocol to support authentication. This solution works well if all DHCP servers and clients would be upgraded to support the new authenticated protocol. However, most of organizations and companies still use unauthenticated DHCP server. This will cause deficiency in term of security and network disruption. In order to use the existing DHCP server and support authentication, a new idea has been suggested in this section. Figure 2 shows the proposed method for Securing WLAN against Rogue DHCP server and unauthorized clients with static IP

A client configured to get a dynamic IP address from the DHCP server goes through 4 steps as shown in Figure 2 (step1 to step 4, or step 7 to 10). However, following these phases client would try to authenticate with the DHCP server by passing authentication request as shown in step 5 or 11. This authentication request would be done by passing the digital signature of the client along with the message request to the DHCP server, which is encrypted by the public key of a valid DHCP server. The DHCP server would verify this authentication request through the public key of the client if it can decrypt this message as shown in step 12. The public keys of the DHCP server and clients are known to each other. However, if the IP was obtained from a rogue DHCP server, this server would fail to authenticate this request because it was encrypted by another public key of valid DHCP server as shown in step 6. If the IP was obtained from a valid DHCP server, the authentication phase would pass as shown in steps 11 and 12.

To avoid clients from using a valid IP before been authenticated by the DHCP server, this server at the AP or at standalone terminal would generate temporary IP which will be given to clients as shown in step 10. This temporary IP would allow the clients to communicate to the DHCP server while been authenticated by this server. However, this temporary IP doesn’t allow clients to communicate over the network since it is based on different subnet and different server’s configuration. If these clients passed the authentication phases shown in step 11 and 12, it would be given a permanent IP by the server. This permanent IP would allow the clients to use the network and access its resources. This mechanism of authenticating the clients following the dynamic IP assignment, would allow client detecting a rogue DHCP server connected to the network. It also allows the DHCP server to deny any clients that are not registered as official client. This will prevent unauthorized clients from getting a dynamic IP and accessing the network resources. These unauthorized clients would have public key and MAC address not known to the DHCP server.

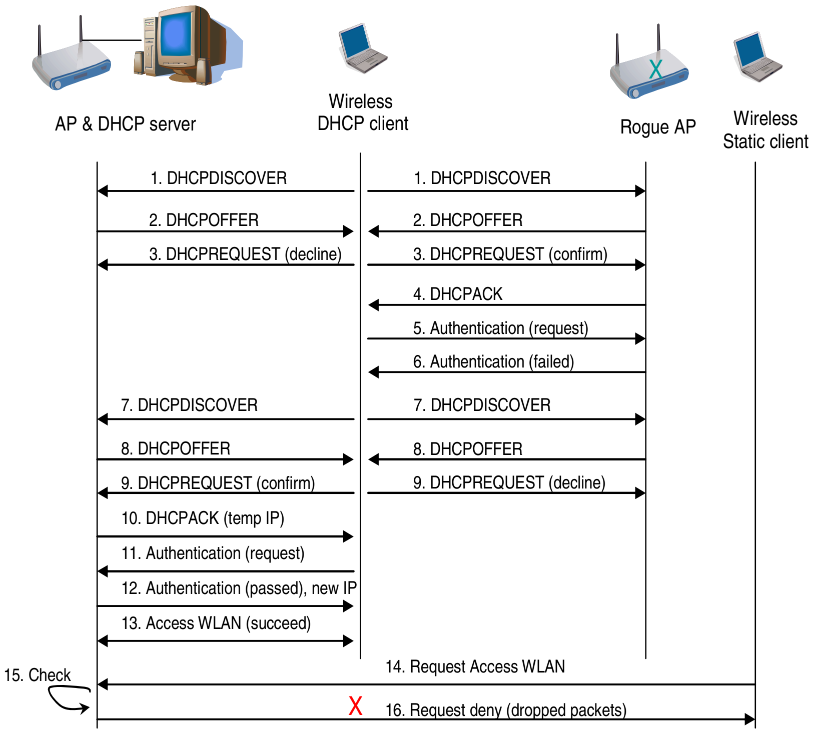


Fig. 2. Securing WLAN against Rogue DHCP server and un-authorized clients with static IP

The above suggested solution is based on new technique for providing security over existing DHCP server. This will require minimum changes to the existing DHCP server and client compared to authenticated DHCP server proposed by IEEE. The complexity of adding extra step to secure the existing DHCP would still use the existing software of the DHCP without the need to change its protocol. While Authenticated DHCP requires building new protocol and must replace the existing DHCP in order provide security for the network. Therefore, there will be an enormous effort and cost to implement some changes in the existing DHCP to support authentication while using our method operating systems using exiting DHCP server would benefit from supporting a secure DHCP server against rogue one. In addition, in spite of the suggestion of authenticated DHCP server protocol few years ago, most of the servers; windows 2003 and Linux still support the old DHCP protocol. Implementing the suggested technique is still under investigation. Once implementation done, we can compare the performance of both techniques in term of fast authentication and complexity of software development. However, it is clear and logical that both techniques provide security and address the rouge DHCP, but the technique suggested by in this paper would require less complexity to develop it and minimum changes to existing network infrastructure that used existing DHCP server.

2.3 Securing the Network Against Un-authorized Clients with Static IPs

In addition to securing the network against rogue DHCP server as explained in section 2.2, the authentication mechanism would be used to prevent unauthorized clients from accessing the WLAN. Unauthorized client can be any client who has a static IP that makes it able to access the network, or one obtained a dynamic IP from rogue DHCP server based on the same subnet. To prevent these unauthorized clients from accessing the network through wireless AP or through switch, their packets would be analyzed at the AP or switch against the authenticated list held at the DHCP server. This is done by comparing their IP and MAC addresses to the list of registered clients at the DHCP server. Since this list contain a valid client who is authorized and registered through the DHCP server, packets coming from other clients not included in the list will be dropped by the AP or switch connected to AP as shown in steps 14, 15, and 15 of figure 2.

3 Conclusion

The new architecture described in Figure 1, and 2 has the potential to improve the performance, and deployment effectiveness of enterprise and other large-scale wireless LANs. This has been done through securing the distributed wireless management. A new security mechanism used to detect a rogue AP and stop it from joining the corporate WLAN. Furthermore, a security mechanism has been suggested to detect a rogue AP as well as rogue bridge AP. This will improve the corporate network security and stop illegal access of rogue mobile terminals that try to benefit from the corporate network and have the chance of accessing its network and hack its resources. In addition, a new mechanism of mutual authentication between the DHCP server and clients has been described. This will allow clients to detect a rogue DHCP server and deny its IP assignment. It also allows the DHCP server to detect unauthorized clients and deny their requests of obtaining dynamic IP and used them to access the network resources. Implementing the new suggested technique to detect a rogue DHCP server is still under investigation. Then the performance of the suggested technique in term of complexity and speed of authentication would be compared to authenticated DHCP presented by [10]. The proposed solution would allow organization keeping the existing DHCP server while it provides a secure system with minimum configuration and at lower cost. Moreover, it allows the AP or switches to synchronize with the DHCP server and drops any packet generated by the unauthorized Wireless clients.

References

1. IEEE 802.11, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications (1997)
2. Kyriazakos, S., Karestos, G.: Practcial Resource Management In Wireless Systems. Book Reviews/Edited by Andrzej Jajszczyk IEEE Communications Magazine 42(11), 12–14 (2004)

3. Ush-Shamszaman, Z., Samiul Bashar, Md., Abdur Razzaque, M., Showkat Ara, S., Sumi, J.K.: A Mobility Management Scheme in All-IP Integrated Network, A Mobility Management Scheme in All-IP Integrated Network. In: Proceedings of the 23rd IASTED International Multi-Conference Parallel and Distributed Computing And Networks, Innsbruck, Austria, pp. 32–37 (February 2005)
4. DaSilva, L.A., Midkiff, S.F., Park, J.S., Hadjichristofi, G.C., Davis, N.J., Phanse, K.S.: Tao Lin Network Mobility and Protocol Interoperability in Ad Hoc Networks. IEEE Communications Magazine 42(11), 88–96 (2004)
5. Kbar, G., Mansoor, W.: Distributed Resource Management in Wireless LANs. International Journal of Business Data Communications and Networking 2(4), 47–59 (2006)
6. Proxim Wireless network, Rogue Access Point Detection: Automatically Detect and Manage Wireless Threats to Your Network, White paper TechRepublic, CNET Networks (May 2004), http://www.proxim.com/learn/library/whitepapers/Rogue_Access_Point_Detection.pdf,
7. Henders, R., Opdyke, B.: Detecting Intruders on a Campus Network: Might the Threat Be Coming From Within? In: ACM SIGUCCS 2005, Monterey, California, USA, pp. 113–117 (November 2005)
8. Sommer, R., Paxson, V.: Enhancing Byte-Level Network Intrusion Detection Signatures with Context. In: CCS 2003, pp. 262–271. ACM, Washington, DC, USA (2003)
9. Zhang, Y., Lee, W., Huang, Y.: Intrusion Detection Techniques for MobileWireless Networks. Wireless Networks 9(5), 545–556 (2003)
10. Droms, R., Arbaugh, W.: Authentication for DHCP Messages, RFC 3118, Internet Engineering Task Force (June 2001) [9.2.2004], http://ww.ietf.org/internet-drafts/draft-ietf-dhc-dhcpv6-staeless-04.txt
11. Outi Annala, The Hot Topics in DHCP Protocol Development, www.cs.helsinki.fi/u/kraatika/Courses/IPsem04s/DHCP_HotTopics.pdf
12. Spácl, R., Ikonen, J., Porras, J.: Forcing Usage Rules in Public Wireless LANs. In: Proceedings of the 27th Annual IEEE Conference on Local Computer Networks, Tampa, Florida, USA, p. 0415 (November 2002)

Structures for Communication-Efficient Public Key Revocation in Ubiquitous Sensor Network*

Abedelaziz Mohaisen[1], DaeHun Nyang[1,**], YoungJae Maeng[1], and KyungHee Lee[2]

[1] Information Security Research Laboratory, Inha University
Incheon 402-751, Korea
asm@seclab.inha.ac.kr, nyang@inha.ac.kr,
brendig@seclab.inha.ac.kr

[2] Department of Electrical Engineering, The University of Suwon
Suwon 445-743, Korea
khlee@suwon.ac.kr

Abstract. In this paper we discuss the uprising problem of public key revocation. The main problem in key revocation includes the relatively large memory and communication required to store and transmit the revoked list of keys. This problem becomes serious as the sensor network is subjected to several constraints. In this paper, we introduce several efficient representation mechanisms for representing a set of revoked identifiers of keys. We discuss several network and revocation scenarios and introduce the corresponding solution for each. To demonstrate the value of our proposed approaches, practical simulation results and several comparisons with the current used revocation mechanism are included.

Keywords: Key revocation, sensor network, communication efficiency, complete subtree cover, bit vector, dynamic run-length encoding.

1 Introduction

The security of wireless sensor network (WSN) has been the issue of critical and challenging research. Due to the limited resources of WSN, public key (PK) algorithms has been discarded from being a solution due to their known computational requirements that enables vulnerability for DoS attacks [1]. In lieu, symmetric key (SK) algorithms have been studied where several SK-related problems have been researched. Because of the weak infrastructure of the WSN that does not permit any traditional symmetric key distribution mechanism such like the key distribution center (KDC), the key pre-distribution (KPD) in which keys or keying material are assigned for each sensor node in a pre-deployment phase has been studied and several schemes have been introduced [2,3,4,5]. However, the recent advancement of operating public key algorithms such like

* 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 Advancement)(IITA-2006-C1090-0603-0028).

** Corresponding author.

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 822–833, 2007.

RSA [6] and ECC [7] on typical sensor nodes has shown a great feasibility that promises a good solution for many of the current security problems in sensor network [8,9,10,11]. Once PK algorithms are deployed in WSN, both of the resiliency and connectivity problems will not be a problem any more [12,13]. However, to make PK algorithms more efficient, different key management services will be required. This mainly includes PK authentication and revocation. As the the cost of communication is much greater than the cost of the local computation on sensor nodes, any efficient security service of PK in WSN should be as light as possible [14]. In this paper, we investigate the problem of the second issue (i.e., PK revocation) and introduce communication efficient schemes to solve this problem.

In the traditional networks, digital certificates (DCs) are used for the purpose of key distribution and authentication. In addition, the certificate revocation is used to revoke a certain certificate for any reason. The digital certificate (e.g. X.509 [15]) includes certificate identifier or simply serial number, *public key which is 1024 bits in case of RSA*, certificate attributes, and digital signature of the certificate's contents. To revoke a PK which is associated with a certificate, the identifier that represents the undesired PK's certificate is published through the entire network. If the number of certificates to be revoked is more than one certificate, their IDs are listed in a Certificate Revocation List (CRL). In the CRL, the IDs of the certificates associated with PKs to be revoked are represented as in Eq. (1) considering the definition of naïve representation in 1. Obviously, the resulting overhead of such representation increases sharply as the number IDs to be revoked increase.

In WSN, we assume that both of the keys and their associated certificates which are for a specific node hold the node's ID. Through the rest of this paper, PK and DC (or digital certificate) are used interchangeably to refer to the revoked identifier.

2 Contributions and Organization

In this paper, we investigate efficient communication schemes for DC revocation and consider the naïve representation scheme defined in 1 as reference work with which our work is compared. Our contribution thereafter includes three parts: (i) introducing a scheme based on the complete subtree cover among the subset coverage framework to represent a set of certificates' identifiers. (ii) Introducing a novel Bit Vector Scheme (BVS) for DC revocation. We study the case of dynamic and static revocation approaches which are equivalent to multiplies and single revocation for same entity (i.e., PK or DC). (iii) Study the probability of specific pattern occurrence in the revocation list. To eliminate extra overhead, we introduce a dynamic run length encoding algorithm that uses pattern-based parameters for efficient compression. In order to demonstrate the performance of the proposed schemes, we introduce a detailed analysis and a practical simulation followed by an extensive comparison with the related works.

The rest of this paper is orgainzed as follows: in section 3, we list the related works on public key and its revocation in sensor networks. In section 4 we introduce the definitions used through the paper. For the details of our contribution, we discuss the naïve and subtree cover mechanisms in section 5. In section 6 we introduce the bit vector scheme with different revocation scenarios. For the efficiency of bit vector compression, we introduce the dynamic run length encoding scheme in 7. We discuss the simulation results in section 8 followed by concluding remarks in section 9.

3 Related Works

The recent results of operating PK algorithms on typical sensor nodes have shown very relevant computational efficiency on contrast of what has been considered as impractical operation for long time. For example, in Gura's *et al.* work [8], practical measurements for elliptic curve cryptography (ECC) [7] and RSA[6] signatures verification have been obtained. It was shown that the verification of ECC signature consumes 1.62 seconds on the typical 8-bit ATmega128 processor which operates at 8 Mega Hertz. In addition, a reduction in the required memory for the code and the data implementation of these algorithms is introduced on other processing platforms (i.e., CC1010 that operates at 14.7456 Mega Hertz). As an extension, Gura *et al.* considered a better implementation of the above algorithms for more efficient energy consumption in [10]. Also, Watro *et al.* developed another limited PK architecture with a practical evaluation of consumed resources per sensor node in so what called TinyPK [11]. The key distribution in TinyOS [16] based on ECC [7] with real measurement and evaluation was taken into account in Malan's *et al.* work [9]. Most recently (February 2007), Ning *et al.* released an update for TinyECC which is an efficient implementation of ECC on TinyOS considering several WSN platforms including MICAz, TelosB and Imote2 motes [17]. To provide PK services such like authentication, Du *et al.* studied the usage of Merkle Authentication Trees [18] and the deployment knowledge to authenticate the public key with memory/communication trade-offs [12]. Also, Nyang *et al.* provided a MAC-based cooperative public key authentication with trade-offs in provided security level and required resources [13].

To the best of our knowledge, PK revocation in WSN has not been studied yet. Thus, our work is the first that formulates the problem of communication efficiency of PK revocation and introduces solutions for this problem.

4 Definitions

In this section, we introduce four definitions which our work is mainly based on. For the Complete Subtree cover (CS) [19,20], we only use the reduction method of identifiers representation and do not use the key generation or assignment mechanisms. In CS, we consider a complete binary tree $\mathcal{T}$ with leaves that represent the different network nodes' DCs. These nodes are represented as $\mathcal{N} : |\mathcal{N}| = n$. The different path from the root to the leaf represents a corresponding DC's ID. The path itself in the tree is represented as left branch of zero and right branch of one relatively from the corresponding parent. The set of node's identifiers to be revoked is $\mathcal{R} = \{v_1, v_2 \dots v_r\} : |\mathcal{R}| = r$. In definition 2, we are interested in the COVER which is the reduced representation for the set of IDs as we will show in 5.

Definition 1 (Naïve Representation Method). *For a space S that permits $2^{|S|}$ possible IDs representation, the naïve method for the identifiers representation in performed by listing these identifiers at same length resulting that the size of the list is the* *length of the list multiplied by S. The general representation of this method is shown in Eq. (1).*

Definition 2 (Complete Subtree Cover). *The CS cover for a group of IDs associated with $\mathcal{R} \subset \mathcal{T}$ is obtained by finding the set of nodes $V_1 \dots V_t \subset \mathcal{T}$ such that $\{v_{i1}, v_{i2} \dots v_{ix}\}$ that represent a complete subtree are rooted at V_i for each i such that $0 < i < t$. Also, $\overline{V_i} \cap \overline{V_j} = \phi$ for any $i \neq j$ and $\overline{V_1} \cup \overline{V_2} \dots \cup \overline{V_t} = \mathcal{R}$ considering that $\overline{V_i}$ is a representative group for the set of nodes rooted at V_i. Here, cover ID for nodes rooted at V_i is the binary string constructed by concatenating bits assigned to each branch from the root to the V_i. Note that the length of ID is always less than or equal to $\lg n$.*

Definition 3 (Bit Vector Scheme). *The Bit Vector Scheme is a relative representation mechanism for sequential identifiers representation aimed to reduce the length of the CRL. In BVS, for a network of n nodes, a bit vector $\mathcal{S}$ n bits is constructed. In $\mathcal{S}$, the i^{th} bit indicates the validity of a key associated with the identifier i. Also, a '0' valued location in $\mathcal{S}$ represents valid or unrevoked PK and '1' represents the revoked PK. In the node's side, keys identifiers to be revoked can be extracted from offset of the 1's occurrence.*

Definition 4 (Dynamic Run-Length Encoding). *The run length encoding (a.k.a. RLE [21]) is a data encoding algorithm that maps a plain binary string $\mathcal{P} : \mathcal{P} = p_1 p_2 p_3 \dots p_a$ into an encoded binary string $\mathcal{C} : \mathcal{C} = c_1 c_2 c_3 \dots c_b$. Considering the instances $\xi_i, \xi_j \in \mathcal{C}$ and $\Psi_i, \Psi_j \in \mathcal{P}$, $\xi_i = \xi_j \iff \Psi_i = \Psi_j$. The notation hRLE-$l$ includes both h and l as the header and the length that determines the prosperities of plain word W_p of length $\|W_p\|$ and the coded word W_c of length $\|W_c\|$ as follows: (a) $MAX(\|W_p\|) = 2^{l-1}$. (b) $MAX(\|W_c\|) = l$. (c) header bit of $W_p = h$. Further examples on this mapping with different h and l are shown in Fig. 2(a), Fig. 2(b), and Fig. 2(c).*

5 Naïve Versus CS Cover for Revoked ID Representation

In the CRL, if the number of the DCs to be revoked is r in a network of n nodes, the required overhead therein is $C = r \times \lceil \log n \rceil$ bits excluding other information such as signature and the CRL's attributes. In this section, we introduce the CS Cover-based schemes to reduce the representation length of the CRL.

5.1 Naïve Representation

Based on Definition 1, for a set of nodes $\mathcal{R}$ of size r in a network of n nodes, the required representation for these identifiers is $r \log_2 n$ bits. In other word, once these identifiers are to be revoked, the message in Eq. (1) is to be sent. In Eq. (1), $b_{(i)(j)}$ represents the i^{th} bit in the j^{th} revoked identifier representation.

$$\text{RL} = \left\{ \begin{matrix} b_{(1)(k_1)} & b_{(2)(k_1)} & b_{(3)(k_1)} & \cdots & b_{(n-1)(k_1)} & b_{(n)(k_1)} \\ b_{(1)(k_2)} & b_{(2)(k_2)} & b_{(3)(k_2)} & \cdots & b_{(n-1)(k_2)} & b_{(n)(k_2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ b_{(1)(k_r)} & b_{(2)(k_r)} & b_{(3)(k_r)} & \cdots & b_{(n-1)(k_r)} & b_{(n)(k_r)} \end{matrix} \right\} \qquad (1)$$

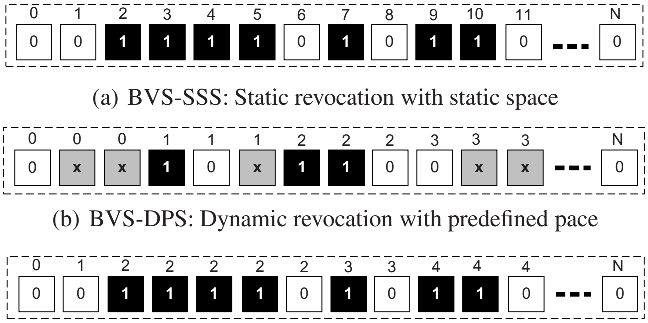
5.2 Complete Subtree in DC Revocation

The complete subtree cover concept in Definition 2 can be used directly to reduce the representation size for the list of the key's to be revoked. As an example, if the set of identifiers $\mathcal{R}$ to be revoked is $\{0100, 0101, 0110, 0111, 1010, 1011$ and $1111\}$. For the first four IDs, the furthest common parent which includes all of those IDs and no other leaf IDs is 01. For the next two, it is 101 and for the last ID, it is 1111 since it has no neighbored IDs to be revoked. Thus, The final list representation is $\{01{,}101{,}1111\}$ which includes all of the required IDs. From the simulation, CS probably reduce the overhead when r is about 5% of n but it greatly reduces the overhead when r is large enough (say, $r > 20\%$ of n). This efficeincy of reduction is shown in Fig. 5(d). Technically, additional overhead equivelent to $\log_2 n * \log_2(\log_2 n)$ is required for shortened covers' separation.

6 Bit Vector Scheme for Efficient DC Revocation

6.1 Motivation: Static Revocation with Static Space (BVS-SSS)

The bit vector which is a relative representation mechanism for the different DCs' IDs including those to be revoked can be used for CRL length reduction. In BVS, as in definition in 3, for the network of n nodes, a bit vector of length n bits in which the $i^{\mathbf{th}}$ bit indicates the validity of the DC of the sensor node with ID i is generated. For the bits of the bit vector, '0' valued location represents unrevoked DC and '1' represents the revoked DC as shown in Fig. 1(a) and the revocation list (RL) expressed in Eq. (2). Since each DC has only one revocation chance with predefined representation space, we call this scenario as the Static with Static Space (BVS-SSS).


0 1 2 3 4 5 6 7 8 9 10 11 N
0 0 1 1 1 1 0 1 0 1 1 0 --- 0
0 0 0 1 1 1 2 2 2 3 3 3 N
0 x x 1 0 x 1 1 0 0 x x --- 0
0 1 2 2 2 2 2 3 3 4 4 4 N
0 0 1 1 1 1 0 1 0 1 1 0 --- 0

(a) BVS-SSS: Static revocation with static space

(b) BVS-DPS: Dynamic revocation with predefined pace

(c) BVS-DDS: Dynamic revocation with dynamic space

Fig. 1. Different scenarios of BVS

Obviously, in the actual DC revocation systems, if a DC is revoked another DC with other ID will replace it. Therefore, in the following, we extend the BVS-SSS to cover the multi-revocation using two approaches.

$$\text{RL} = b_{(0)}, b_{(1)}, \ldots, b_{(N-2)}, b_{(N-1)}. : b_{(i)} = \begin{cases} 1 & \text{If } i \text{ is revoked} \\ 0 & \text{Otherwise} \end{cases} \tag{2}$$

6.2 Dynamic Revocation with Steady Predefined Space (BVS-DPS)

In this approach, we provide each sensor node's DC with a predefined number of bits (BVS-DPS) to handle a number of revocations. For each node, this number is the same to provide an auto-separation mechanism. This procedure provide a better solution than the naïve, however, the representation has a low efficiency when the number of revoked DCs is small.

Let $b_{(i)(j)}$ be the j^{th} revocation chance for the i^{th} node, the RL general representation is shown in Eq. (3) where each bit has three possible statuses. An illustrating example is shown in Fig. 1(b).

$$\text{RL} = \left\{ \begin{matrix} b_{(0)(1)} & b_{(0)(2)} & \cdots & b_{(0)(c)} \\ b_{(1)(1)} & b_{(1)(2)} & \cdots & b_{(1)(c)} \\ \vdots & \vdots & \ddots & \vdots \\ b_{(n-1)(1)} & b_{(n-1)(2)} & \cdots & b_{(n-1)(c)} \end{matrix} \right\} : b_{(i)(j)} = \begin{cases} 0 & \text{if revoked} \\ 1 & \text{if unrevoked} \\ \text{x} & \text{if unused} \end{cases} \tag{3}$$

6.3 Dynamic Revocation with Dynamically Extendable Space (BVS-DDS)

To overcome the efficiency problem of the above representation, let us consider the BVS-SSS with a slight modification. Initially, the bit vector is initialized by '0's when no DC is revoked. Once a DC is revoked, a '1' is attached immediately before the corresponding DC's '0' offset. Based on that, one revocation adds only one bit to the bit vector. When we would like to check the validity of the i^{th} DC, firstly we find the i^{th} block that represent sensor node i and count the number of '1's which means that the $(n+1)^{\text{th}}$ DC for node i is valid. The i^{th} block for node i is composed of the i^{th} '0' in the bit vector and previous '1's before the $(i-1)^{\text{th}}$ '0'. As in the first scenario, each node can have a limited and pre-defined number of revocation chances. An example that shows how this scheme works is in Fig. 1(c).

Unlike other scenarios (i.e., BVS-SSS, BVS-DPS), BVS-DDS is fully dynamic in that it can support a infinite times of revocations where the CRL cost is typically as much as the number of revoked DCs added to an initial overhead. To introduce a dynamic naïve revocation scheme, a larger space to represent IDs is required. While 14 bits are enough to represent 10,000 DC for 10,000 with their associated PKs, 18 bits are required to provide 10 revocation chances for the same size. In BVS-DPS, an equivalent number of bits per node is required.

7 Compression: Dynamic RLE with Predefined Parameters

Technically, knowledge or even probable knowledge represented in probability of the presence for a specific pattern in the RL in BVS-SSS, BVS-DPS, BVS-DDS, CS makes it possible to use the encoding mechanism (RLE) in [4] for an efficient compression. The compression efficiency is due to a long $W_p \in \mathcal{P}$ which are replaced with shorter $W_p \in \mathcal{C}$.

W_p	W_c	$P_r(W_p)$	W_p	W_c	$P_r(W_p)$
1	0000	$p_0^0p_1^1$	000001	0101	$p_0^5p_1^1$
01	0001	$p_0^1p_1^1$	0000001	0110	$p_0^6p_1^1$
001	0010	$p_0^2p_1^1$	00000001	0111	$p_0^7p_1^1$
0001	0011	$p_0^3p_1^1$	00000000	1	$p_0^8p_1^0$
00001	0100	$p_0^4p_1^1$			

(a)

W_p	W_c	$P_r(W_p)$	W_p	W_c	$P_r(W_p)$
0	0000	$p_0^1p_1^0$	111110	0101	$p_0^1p_1^5$
10	0001	$p_0^1p_1^1$	1111110	0110	$p_0^1p_1^6$
110	0010	$p_0^1p_1^2$	11111110	0111	$p_0^1p_1^7$
1110	0011	$p_0^1p_1^3$	11111111	1000	$p_0^0p_1^8$
11110	0100	$p_0^1p_1^4$			

(b)

W_p	W_c	$P_r(W_p)$	W_p	W_c	$P_r(W_p)$
1	00	$p_0^0p_1^1$	0	00	$p_0^1p_1^0$
01	01	$p_0^1p_1^1$	10	01	$p_0^1p_1^1$
00	1	$p_0^2p_1^0$	11	1	$p_0^0p_1^2$

(c)

Fig. 2. (a) RLE with $h = 1$ and $l = 4$ (1RLE4) (b) RLE with $h = 0$ and $l = 4$ (0RLE4) (c) RLE with $h = 1, h = 0$ and $l = 2$ (1RLE4, 0RLE4)

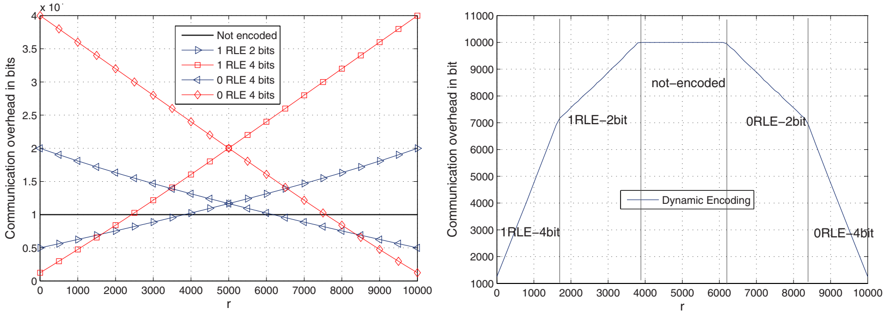
7.1 Parameters Assignment

To get a desirable performance that expresses a correlated efficiency with the number of revoked nodes r, we apply a dynamic encoding by changing l and h in the encoding algorithm. For the relatively small number of revoked DCs, both 1RLE-4b and 1RLE-2b can be used since it is high probability for long consequent zeros to appear. Similarly, for a relatively large number of revoked DCs, 0RLE-2b, 0RLE-4b can be used. Finally, for the case where the number of revoked DCs is similar to the number of unrevoked ones, it is more efficient to keep the RL not encoded. To find out the exact percents where the dynamic encoding parameters should be changed, we encode the same string represent different r as percent of n using the different possible parameters and manually calibrate the points. Fig. 3(a) shows the encoded string using different parameters where the intersections that relies below N is used. Fig. 3(b) shows the RLE for dynamically assigned percentages.

7.2 Encoding Efficiency

To find out the performance of the encoding, the probability of 1s and 0s occurrence need to be measured. Herein, we describe the analysis of efficiency for the BVS-SSS where the other two schemes follow the same analysis.

Let X be a random variable (which is in fact a Bernoulli Random Variable - BRV [22]) that describes the above occurrence of 1 and 0 in $\mathcal{S}$ that represents the revocation bit vector. Based on the above notation and structure of BVS-SSS in Fig. 1(a) and Eq. (2), the probability p_0 is defined as $P_r(X = 0)$ and the probability p_1 is defined as $P_r(X = 1)$. Both of these probabilities are shown in Eq. (4).



(a) RLE with different parameters

(b) Dynamic RLE

Fig. 3. (a) Different simulation running averages for the CS versus the naïve scheme (b) Dynamic RLE with dynamically chosen parameters

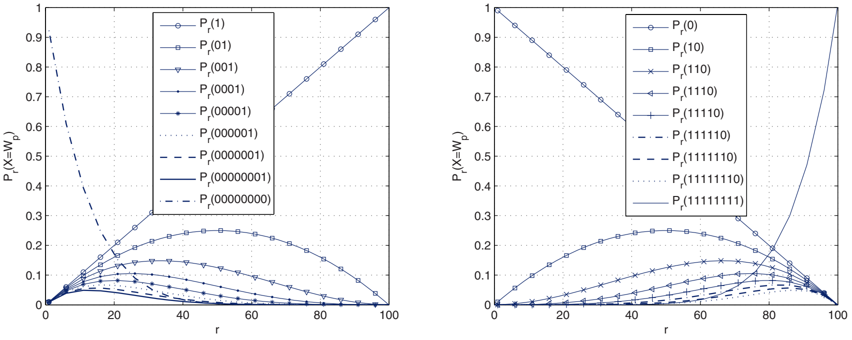
$$P_r(X=0) = p_0 = \left(\frac{n-r}{n}\right),\ P_r(X=1) = p_1 = 1 - p_0 = \left(\frac{r}{n}\right) \qquad (4)$$

Note that, always $p_0 + p_1 = 1$, $0 \leq p_0 \leq 1$, and $0 \leq p_1 \leq 1$ for any n and r. In addition, any sample point with k number of 1s and $(\|W_p\| - k)$ 0s is assigned a probability $p(k, \|W_p\| - k)$ as in 5

$$p(k, \|W_p\| - k) = p_0^{(\|W_p\|-k)} p_1^k \qquad (5)$$

Based on the mapping of the RLE defined earlier and the above probabilities p_0, p_1, and $p(k, \|W_p\| - k)$, the following probability samples: $P_r(X_i = 1) = (\frac{n-r}{n})^0(\frac{r}{n})^1 = p_0^0p_1^1$, $P_r(X_i, X_{i+1} = 01) = (\frac{n-r}{n})^1(\frac{r}{n})^1 = p_0^1p_1^1$, $P_r(X_i, \dots, X_{i+2} = 001) = (\frac{n-r}{n})^2(\frac{r}{n})^1 = p_0^2p_1^1$, $\ddots$, and $P_r(X_i, \dots, X_{i+7} = 00000000) = (\frac{n-r}{n})^8(\frac{r}{n})^0 = p_0^8p_1^0$ where i is the offset in $\mathcal{S}$ for the beginning of W_p. The above are for 1RLE-4 where other RLE encoding follows the same probability representation. From the above family of probabilities, a conclusive representation of any pattern occurrence probability is shown in Fig. 2(a), Fig. 2(b), and Fig. 2(c) for the corresponding pattern with the given W_p, k, l, and h. The resulting probability for some pattern occurrence that demonstrates the efficiency of the encoding is shown in Fig. 4(a) and Fig. 4(b). The overall performance of the dynamic encoding is dependent upon the variation of the parameters (h, l) which always guarantees a high probability for a desirable long and compressible pattern to occur.

Finally, the entropy $H(X)$ of $\mathcal{S}$ per symbol is shown in Eq. (7) which is typically a binary entropy function (as X that describes $\mathcal{S}$ is a Bernoulli Random Variable) resulting that the required bits per symbol in $\mathcal{S}$ are always less than or equal to 1. That means, based on r, there exist an algorithm (as shown in the simulation results) that is able to reduce the length of the bit vector into a shorter compressed one with an efficiency correlated with $H(X)$ in 8.



(a) $P_r(X = W_p)$ in 1RLE-4

(b) $P_r(X = W_p)$ in 0RLE-4

Fig. 4. Probability of occurrence for some pattern that determines the efficiency of encoding with the specified parameters under r number of revoked positions in $\mathcal{S}$ for a sample of $n = 100$

$$H(X) \triangleq -\sum_{i=0}^{1} P_r(X = i)\log_2 P_r(X = i) \tag{6}$$

$$= -\left[\left(\frac{n-r}{n}\right)\log_2\left(\frac{n-r}{n}\right) + \left(\frac{r}{n}\right)\log_2\left(\frac{r}{n}\right)\right] \leq 1 \tag{7}$$

$$= [(e-1)\log_2(1-e) - e\log_2 e] \text{ for some } e : 0 < e \leq 1, e = r/n \tag{8}$$

8 Simulation Results and Analysis

In this section, we justify our schemes' performance by simulating the following schemes: naïve, naïve Encoded, CS, BVS-SSS, BVS-DDS, BVS-DPS. To handle the randomness of the revoked DC ID, we use a random identifier selector that indicates the current compromised ID from the non-compromised pool. $n = 10,000$ sensor nodes and $c = 10$.

Simulation results: The dynamic encoding altering points to change the encoding parameters are calibrated using the intersections of the RLE encoding using different length and heading parameters as of Fig. 3(a) and Fig. 3(b). The communication overhead of the BVS-DDS versus the naïve is shown in Fig. 5(a) and Fig. 5(b). Note that, our BVS-DDS scheme provides a high efficiency since it does not include any non-required bits. Fig. 5(c) shows the resulting performance of the CS in which we performed the simulation on different random samples (i.e., 5 for each) and considered the average (AVG $= \frac{1}{5}\sum_{i=1}^{5}$ CS Overhead$_i$). To show the random behavior of the CS, we executed our simulator for the three times in which the average of 5 times is considered. Note that the performance of the CS provides a higher efficiency when r is large enough.

To show the regions in which the naïve solution provides a better performance than the BVS (i.e. $r \leq \frac{n'}{\lg n - 1}$ in case of BVS-DPS and BVS-DDS), we executed the simulator for different network sizes. Fig. 5(d) shows that the BVS provides a better performance than the naïve when r is greater than 7% of n. Moreover, additional numerical results are shown in Table 1.

Table 1. Communication overhead in bit for revoking different percents of network size using different schemes. $N = 10000$ nodes, $-_{\mathrm{C}}$ indicates the usage of RLE.

Scheme	01%	05%	10%	20%	40%	50%
Naïve	1,400	7,000	14,000	28,000	56,000	70,000
Naïve$_{\mathrm{C}}$	2,595	12,876	26,132	52,104	103,403	130,070
CS	1,355	6,805	13,310	25,004	42,850	49,408
Naïve-DPS	1,800	9,000	18,000	36,000	72,000	90,000
Naïve-DDS	18,000	90,000	180,000	360,000	720,000	900,000
BVS-SSS	10,000	10,000	10,000	10,000	10,000	10,000
BVS-SSS$_{\mathrm{C}}$	1,597	2,994	4,753	7,569	10,000	10,000
BVS-DPS	10,100	10,500	11,000	12,000	14,000	15,000
BVS-DDS	11,000	15,000	20,000	30,000	50,000	60,000
BVS-DDS$_{\mathrm{C}}$	3,756	13,778	20,000	27,508	37,498	41,320

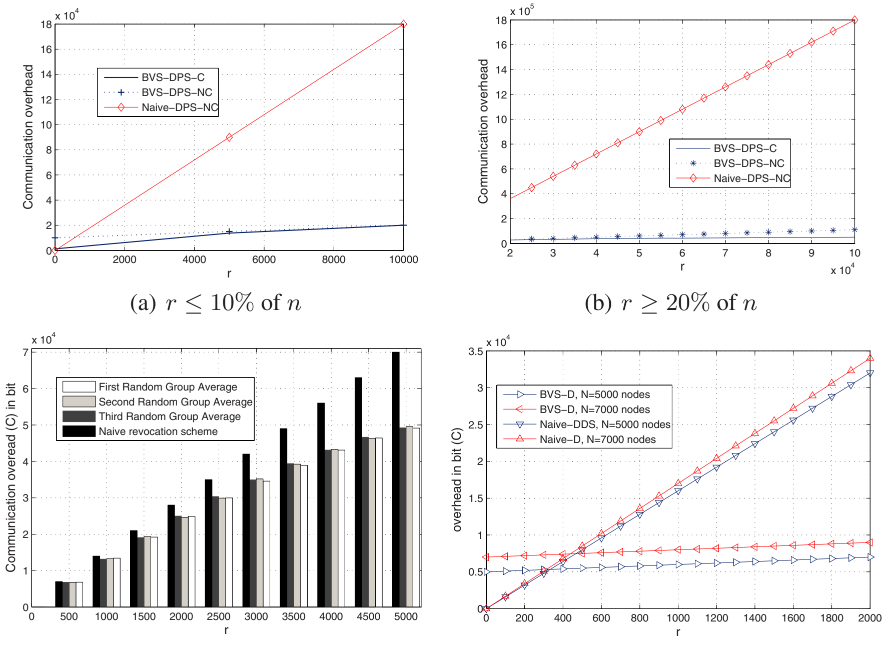
Analysis: For the part of the BVS-SSS, BVS-DPS, and BVS-DDS, the dynamic encoding algorithm with h, l parameters on intervals provides a high efficiency based the early discussed probability of pattern occurrence. On the other hand, due to the non-systematic occurrence of similar bits in both naïve and CS algorithm, the RLE may not provide a high efficiency. That means, applying the dynamic RLE for the CS or the naïve will replace a small string in the $\mathcal{P}$ with a longer ones $\mathcal{C}$ with high probability.

Another notable feature is that the CS algorithm provides a probabilistic communication reduction. If the set of DCs to be revoked is in a consequent order with fewer gaps, CS will provide a high representation efficiency and reduction in the overall communication. Otherwise, the communication will be greater. In the worst case, it will be the same like the naïve representation. Table 1 shows a numerical results that consider a uniformly random distribution for the compromised IDs of DCs. The efficiency is dependent on the random manner of the compromising and the used RLE parameters.

For the BVS-DDS, the communication overhead is $f_r = r + n$ bit where n is the real network size which is equivelant to the initial overhead in bits. In the case of Dynamic naïve solution, the required overhead is $f_r = r * \lg n^{'}$ where $n^{'}$ herein is the expanded space that permits a multiple revocation for a given ID as discussed earlier

Table 2. Overhead Comparison

Scheme	**Communication Overhead**	**Scheme**	**Communication overhead**
Naïve	$r\log_2 n$	**CS**	$r\log_2 \frac{n}{r} + (\log_2 n)(\log_2 \log_2 n)$
BVS-SSS	n	**Naïve-DPS**	$r\log_2 n^{'} = r\log_2 cn$
BVS-DPS	$n^{'} = cn$	**Naïve-DDS**	$r\log_2 n^{'} = r\log_2 cn$
BVS-DDS	$n + r$		



(a) $r \leq 10\%$ of n

(b) $r \geq 20\%$ of n

(c) CS versus the naïve scheme

(d) Naïve DPS versus BVS-DPS

Fig. 5. The communication overhead of our schemes with different scenarios

(i.e., $n^{'} = c * n$ where c is the possible revocation chances for a DC). BVS-SSS provides a better efficiency than the naïve scheme for $r \geq \frac{n^{'}}{\lg n^{'} - 1}$ even without any compression. For the CS communication overhead is, at the worst case, $r \log_2 \frac{n}{r}$ bits [20] in addition to the sets separation bits which we discussed earlier. A concluding overhead comparison is shown in Table 2.

9 Conclusion and Future Works

We introduced two schemes for communication efficient DC revocation in WSN. The first one relies on the complete subtree and the second is bit vector scheme. Our solutions showed a relevant reduction in the communication. The CS provides high probability for reduction in normal cases. The upper bound for the BVS is constant and depends initially on the networks size. Using encoding schemes like RLE can be behind a more reduction in the communication overhead. We tried different scenario of dynamic and static BVS. Trying other encoding/compression schemes and finding other applications for the bit vector representation and/or the CS might be future work. The deployment knowledge as a direct communication reduction method for the communication overhead is valuable to be studied.

References

1. Deng, J., Han, R., Mishra, S.: Defending against path-based dos attacks in wireless sensor networks. In: SASN, pp. 89–96 (2005)
2. Chan, H., Perrig, A., Song, D.X.: Random key predistribution schemes for sensor networks. In: IEEE Symposium on Security and Privacy, p. 197 (2003)
3. Du, W., Deng, J., Han, Y.S., Chen, S., Varshney, P.K.: A key management scheme for wireless sensor networks using deployment knowledge. In: INFOCOM (2004)
4. Eschenauer, L., Gligor, V.D.: A key-management scheme for distributed sensor networks. In: ACM CCS, pp. 41–47 (2002)
5. Liu, D., Ning, P.: Establishing pairwise keys in distributed sensor networks. In: ACM CCS, pp. 52–61 (2003)
6. Rivest, R.L., Shamir, A., Adleman, L.M.: A method for obtaining digital signatures and public-key crypto-systems. CACM 26, 96–99 (1983)
7. Koblitz, N., Menezes, A., Vanstone, S.A.: The state of elliptic curve cryptography. Des. Codes Cryptography 19, 173–193 (2000)
8. Gura, N., Patel, A., Wander, A., Eberle, H., Shantz, S.C.: Comparing elliptic curve cryptography and rsa on 8-bit cpus. In: CHES, pp. 119–132 (2004)
9. Malan, D.J., Welsh, M., Smith, M.D.: A public-key infrastructure for key distribution in tinyos based on elliptic curve cryptography. In: First IEEE Int. Conf. on Sensor and Ad Hoc Comm. and Networks, pp. 71–80 (2004)
10. Wander, A., Gura, N., Eberle, H., Gupta, V., Shantz, S.C.: Energy analysis of public-key cryptography for wireless sensor networks. In: PerCom, pp. 324–328 (2005)
11. Watro, R.J., Kong, D., fenCuti, S., Gardiner, C., Lynn, C., Kruus, P.: Tinypk: securing sensor networks with public key technology. In: SASN, pp. 59–64 (2004)
12. Du, W., Wang, R., Ning, P.: An efficient scheme for authenticating public keys in sensor networks. In: MobiHoc, pp. 58–67 (2005)
13. Nyang, D., Mohaisen, A.: Cooperative public key authentication protocol in wireless sensor network. In: Ma, J., Jin, H., Yang, L.T., Tsai, J.J.-P. (eds.) UIC 2006. LNCS, vol. 4159, pp. 864–873. Springer, Heidelberg (2006)
14. Pottie, G.J., Kaiser, W.J.: Wireless integrated network sensors. Commun. ACM 43, 51–58 (2000)
15. Housley, R., Polk, W., Ford, W., Solo, D.: Rfc 3280: Internet x.509 public key infrastructure: Certificate and certificate revocation list (crl) profile (2002)
16. Levis, P., Madden, S., Gay, D., Polastre, J., Szewczyk, R., Woo, A., Brewer, E.A., Culler, D.E.: The emergence of networking abstractions and techniques in tinyos. In: NSDI, pp. 1–14 (2004)
17. Ning, P., An Liu, P.K.: Tinyecc: Elliptic curve cryptography for sensor networks (version 0.3), software package (2007)
18. Merkle, R.C.: Protocols for public key cryptosystems. In: IEEE S&P, pp. 122–134 (1980)
19. Naor, D., Naor, M., Lotspiech, J.: Revocation and tracing schemes for stateless receivers. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 41–62. Springer, Heidelberg (2001)
20. Fiat, A., Naor, M.: Broadcast encryption. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 480–491. Springer, Heidelberg (1994)
21. Golomb, S.W., Peile, R.E., Scholtz, R.A.: Basic Concepts in Information Theory and Coding: The Adventures of Secret Agent 00111. Springer, Heidelberg (1994)
22. Trivedi, K.S.: Probability and Statistics with Reliability, Queuing and Computer Science Applications. John Wiley and Sons Inc, New York, USA (2001)

A Method of Pair-Wise Key Distribution and Management in Distributed Wireless Sensor Networks

Xing Liao[1], Shizhong Xu[1], Sheng Wang[1], and Kaiyu Zhou[2]

[1] University of Electronic Science and Technology of China,
NO.4, Section 2, North Jianshe Road, Chengdu, China
{liaoxing,xsz,wsh_keylab}@uestc.edu.cn
[2] China Telecom Corporation Limited Beijing Research Institute
708 Guanhua, 118 Xinei, Beijing, China
zhouky@ctbri.com.cn

Abstract. Improvements in technology introduce new application for sensor networks. As Mission-critical applications are deployed in distributed wireless sensor networks, security issues arise. They are facing tremendous challenges: wireless communication environment (usually in hostile areas), lack of infrastructure support, inability of predicting network topology and limited resource associated with nodes. Efficient and robust key distribution is important to secure such kind of sensor networks. To address this issue, we propose a simple pair-wise key distribution and management approach. We predistribute an INITIAL_KEY and a PRIVATE_KEY to each sensor. Let each sensor broadcast its encrypted PRIVATE_KEY with limited power. Then every sensor shares a pair-wise key with each of its neighbors. Any pair of adjacent nodes could communicate securely using a common pair-wise key. Node addition and deletion during communication stage are supported, so as to the update of pair-wise keys. Simulation result proves the scalability of our method.

Keywords: Distributed wireless sensor networks, key distribution and management, pair-wise key, scalable.

1 Introduction

A sensor node typically contains a sensing unit, a processing unit, a power unit, a storage unit, and a wireless transmitter / receiver. In most applications, sensor nodes have to be inexpensive and small. They are usually resource-limited, namely limited power and storage, lack of long-distance communication capability and powerful computing capability. Distributed wireless sensor network (DSN) is composed of large number of this kind of nodes. Foreseeable mission-critical DSN will be widely applied. In hostile or disaster areas surveillance, manual deployment is impossible, besides, a fixed infrastructure establishment is not safe. Generally, hundreds of sensor nodes have to be deployed in the target area by randomly scattering. So it's impossible to know the exact position of each sensor node before deployment, so as to the topology of DSN. In addition, sensor nodes may suffer physical or logical attacks. In order to establish a secure DSN, we must safely distribute keys to every sensor node. After that we could use efficient

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 834–844, 2007.

encryption method and hash functions to protect this DSN. Distributing these crucial keys securely is never trivial, we need some strong and efficient key distribution mechanisms.

Sharing a single key in whole network is not a good idea because an adversary could easily obtain the key. On the contrary, if we use distinct keys for every possible links in the network, each node has to store many useless keys. Although this approach is safe enough, its scalability is poor, i.e., it is not fit for large scale network. As a trade-off, we'd better distribute *m* keys to each node(this *m* keys are randomly drawn from a large key-pool of N keys), so that each pair are able to communicate with a probability of p=1-[(N-m)!]^2/[(N-2m)!N!]. This approach sacrifices connectivity while saving the storage. In recent years, many approaches are proposed. In [1], authors mentioned although there are ongoing works [2], [3], [4] to customize public key cryptography and elliptic key cryptography for low-power devices, such approaches are still considered costly due to high processing requirements.

We propose a simple pair-wise key distribution and management approach to establish a secure DSN. This approach only requires sensor nodes have a few memory storage for keys no matter how large the scale of network, and achieves the similar security as other approaches do. Our approach is composed of four parts: (1) key predistribution, (2) key sharing and discovering, (3) adding new sensor node, and (4) key reinforcement. These four parts include the whole process of key management of every sensor node during their whole lifetime. Our analysis indicates that our approach has more flexibility when comparing with the previous methods.

The rest of this paper is organized as follows. After presenting related work in section 2, we introduce our method in detail including all parts mentioned above, and analyze its features in Section 3. Section 4 gives optional encryption and decryption algorithm. Section 5 presents our simulation which demonstrates our analysis, followed by our conclusion in Section 6.

2 Related Work

In DSNs, sensor nodes could use predistributed keys directly, or use keying materials to dynamically generate pair-wise keys. Challenge is to find an efficient way of distributing keys or keying materials to each sensor node prior to deployment.

In probabilistic solutions, key-chains are randomly selected from a key-pool and distributed to sensor nodes. In [5], it proposed a wonderful method. In key setup phase, a large key-pool of N keys and their identities are generated. For each sensor, *m* keys are randomly drawn from the key-pool. These *m* keys and their identities form the key-chain for a sensor node. Thus, probability of key share among two sensor nodes becomes p=1-[(N-m)!]^2/[(N-2m)!N!]. In key discovery phase, each node broadcasts its *m* key-IDs and finds whether there are some neighbors which also have the same key. In order to communicate with those neighbors who don't have a common key, there must be an agent by which could establish a path-key. In [6], a *q*-composite random key predistribution scheme is proposed. Two nodes must have at least *q* common keys if they intend to communicate. Then both nodes can get a common key by translating those *q* common keys using hash function. This scheme has a remarkable advantage: one captured node only exposes the secrets among its neighbors and itself. But it's very difficult to find out an optimal size of the key pool. If the size is too large, the probability of any pair of nodes to have a common key will be too small. On the contrary, if the size is too small, even a small number of captured nodes may induce a large scale key-space exposure.

In deterministic solutions, deterministic processes are used to design the key-pool and the key-chains to provide better key connectivity. In [7], a location-based key establishment method is proposed. It is supposed that each sensor node could be deployed quite near the prescriptive position. Thus each node could only prestore or receive the keys of its nearest neighbor nodes. When the actual position is approximately equal to the prescriptive position, the connectivity of network is guaranteed while each node's storage resource is also greatly saved. However, its requirement about node's position is too rigorous. In [8], all sensor nodes have a common master key *Km*. If two nodes (*i*, *j*) intend to communicate, both of them first send and receive a random nonce value (*RN*) to another. Then both nodes use *Km* and those two random nonce values *RNi*, *RNj* to generate a session key. After that, they can communicate securely by using this session key. However, in this approach, if *Km* is compromised, the whole network may be in danger.

3 Our Approach

3.1 Key Predistribution

Before scattering sensor nodes to the target area, we install an INITIAL_KEY and a PRIVATE_KEY for each sensor node. Where the INITIAL_KEY is uniform throughout the DSN, and the PRIVATE_KET is distinct for each node. We must carefully choose INITIAL_KEY in order to make it very difficult to be decrypted when adversaries intercept or capture the messages encrypted by it. It's impossible to requiring INITIAL_KEY to be safe all the time, what we need is make sure it's safe during the key sharing and discovering period. After that, INITIAL_KEY becomes useless. At the same time, we should promise adversaries couldn't obtain INITIAL_KEY even they captured some sensor nodes before or during the key sharing and discovering period, for instance, we may design sensor nodes like this: use temporary memory to save INITIAL_KEY. When sensor node suffers from physical attack, INITIAL_KEY will be deleted from memory automatically.

3.2 Key Sharing and Discovering

3.2.1 Key Sharing and Discovering Phase

Supposing sensor nodes have already been scattered to the target area. At the beginning, every sensor nodes use INITIAL_KEY to encrypt its PRIVATE_KEY and ID, and broadcast this message using predefined power (We could probably determine the density of network before deploying sensor nodes, or deploy nodes according to a predetermined density requirement). So that the radio wave could only reach some predetermined limited range, and received by a small number of nearest neighbors. These neighbors use common INITIAL_KEY to decrypt the sender's PRIVATE_KEY and ID, and then store them in their permanent memories. At the end of this PRIVATE_KEY discovery phase, all nodes erase their temporary memories.

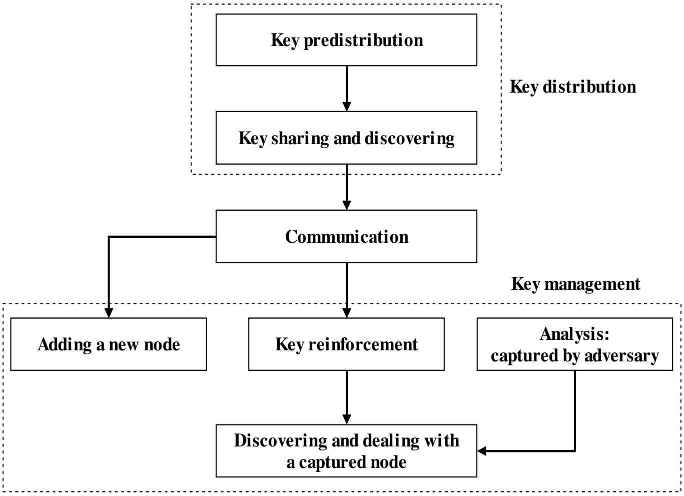

Key predistribution
Key sharing and discovering
Key distribution
Communication
Key management
Adding a new node
Key reinforcement
Analysis: captured by adversary
Discovering and dealing with a captured node

Fig. 1. The overview of our approach. Our method contains two kernels: key distribution and key management. Each part of them has their own features.

We save a lot of the memory space for each sensor, since only the nearest neighbors need to receive and store the key. As a result, each sensor only sends and receives a few messages, stores useful keys. Each sensor node's PRIVATE_KEY is localized in its neighborhood. This property reinforces network's security, which we'll discuss later. In addition, this approach is fit for all networks of any size while providing a good connectivity.

3.2.2 Communication Phase

After key sharing and discovering phase, each sensor node exactly knows its nearest neighbors. If we know the density of network, we can control the power of broadcast to restrict the number of the nearest neighbors. When that's the case, each sensor node can only communicate with its nearest neighbors since it only knows their PRIVATE_KEYs. In this way, we improve the security of network in respect that a captured node couldn't exposure more PRIVATE_KEYs other than those of its nearest neighbors.

In the communication phase, the sender, firstly, encrypts data by using its own PRIVATE_KEY, secondly, adds its ID in front of the message, thirdly, encrypts this message by using receiver's PRIVATE_KEY, finally, sends out this double-encrypted message. When each of its nearest neighbors receives that double-encrypted message, in the beginning, it makes use of the PRIVATE_KEY which originally belong to it (Not the PRIVATE_KEYs received from the others) to decrypt that message. Obviously, only the correct receiver could decrypt it. If success, the receiver reads sender-ID, uses this ID to search the PRIVATE_KEY of the sender from its permanent memory, and then decrypts this message to obtain original data. There is one exception: the receiver couldn't find a PRIVATE_KEY from its memory by using sender-ID. This will happen when we add a new sensor node into the network.

3.3 Adding a New Sensor Node

3.3.1 Motivations and Requirements

In some special circumstances, we probably need to add a new sensor node into the network. We may have a strong purpose, such as to supplement a damaged or exhausted sensor node in a wireless link of communication. Before adding a new node, as a necessary condition, we have to know which node we intend to communicate with and what its PRIVATE_KEY and PRE_PRIVATE_KEY are. For the legitimate sensor node, this is not difficult. The network administrator could securely assign this kind of information to this new node. And he / she should also do his / her possible to make sure that the sensor node which the new one intends to communicate with is secure. It's vital.

3.3.2 Adding a New Sensor Node

The new node uses receiver's original PRIVATE_KEY (When nodes reinforce the PRIVATE_KEY, only this original one must be backuped. We'll discuss key reinforcement later) to encrypt its own PRIVATE_KEY and ID, and then sends this message. After decrypted the message, the receiver knows there is a new node which intend to take part in the network. In order to authenticate this new node, the receiver sends a question to it, asking whether it knows the key named PRE_PRIVATE_KEY of the question-sender. Of course that question is double-encrypted by the PRIVATE_KEY of the receiver and the sender. (Before deployment, under the control of the network administrator, each new sensor node will store the PRE_PRIVATE_KEY of the node which it intend to communicate with. If the new sensor node is captured, it will delete the PRE_PRIVATE_KEY immediately.) The new node answers that by sending PRE_PRIVATE_KEY which is also double-encrypted. In the next, the receiver decrypts that answer and achieves PRE_PRIVATE_KEY. It computes PRE_PRIVATE_KEY by using hash algorithm which has already been stored in its memory since it was produced. If the result is exactly equal to the original PRIVATE_KEY which stored in its memory, the receiver authenticates this new node as a creditable node. This process is showed in Fig.2. Finally, both sides of communication delete that PRE_PRIVATE_KEY completely. To some extent, this insures the security of the PRE_PRIVATE_KEY.

3.3.3 PRE_PRIVATE_KEY and Hash Algorithm

Hash algorithm can make sure that from PRE_PRIVATE_KEY we can easily obtain PRIVATE_KEY, but it's impossible or very difficult to find out PRE_PRIVATE_KEY by computing PRIVATE_KEY. This is the most important characteristic of the hash function. The PRIVATE_KEY which was installed in each sensor node before deployment, in fact, is exactly the result calculated from the PRE_PRIVATE_KEY by hash algorithm. Obviously, the network administrator must keep all the PRIVATE_KEY together with its corresponding PRE_PRIVATE_KEY.

We only offer a reference to add a new sensor node. In fact, this approach requires the new sensor node be precisely deployed at the correct position.

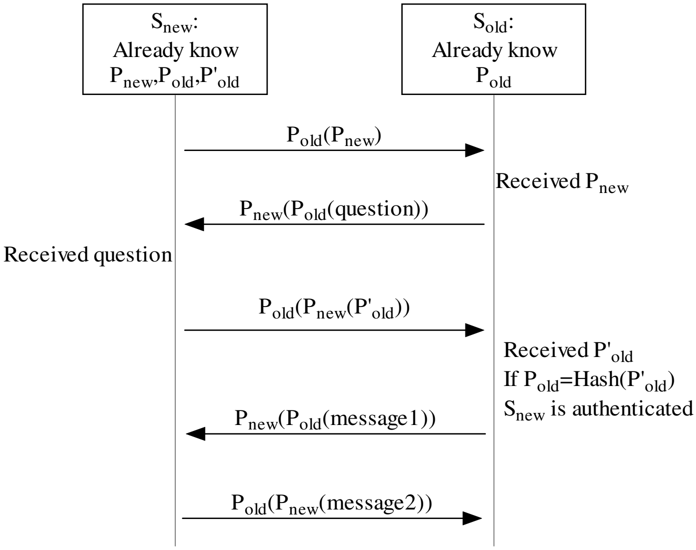

S_{new}:
Already know
P_{new},P_{old},P'_{old}
S_{old}:
Already know
P_{old}
$P_{old}(P_{new})$
Received P_{new}
$P_{new}(P_{old}$(question))
Received question
$P_{old}(P_{new}(P'_{old}))$
Received P'_{old}
If P_{old}=Hash(P'_{old})
S_{new} is authenticated
$P_{new}(P_{old}$(message1))
$P_{old}(P_{new}$(message2))

Fig. 2. The process of adding a new sensor node: S_{new} is the new sensor node, S_{old} is the old one. P_{new} is the PRIVATE_KEY of S_{new}, P_{old} and P'_{old} are the PRIVATE_KEY and PRE_PREIVATE_KEY of S_{old}.

3.4 Analysis: Captured by Adversary

We may design the sensor nodes like this: When suffers from physical attack, it'll continually send the being-captured messages to tell its neighbors till its energy has been exhausted. But this is far from enough. If a node is captured by adversary, all information which it is storing will get revealed, including INITIAL_KEY, its own PRIVATE_KEY, PRIVATE_KEYs of its nearest neighbors, hash algorithm, encryption and decryption algorithm(s), and so on. That means not only the captured node, but also its neighbors are unsafe any more. We can call these unsafe neighbor nodes Victim_Neighbors.

(1) If two Victim_Neighbors couldn't communicate with each other, only the security of communication among the captured node and its neighbors is affected. The neighbors of the captured node could communicate with other nodes all the same.
(2) On the contrary, if two Victim_Neighbors could communicate with each other, the adversary can get messages between these two Victim_Neighbors by eavesdroppping. Even worse, the adversary may add some illegitimate nodes in the network, disguise an illegitimate node to one of the Victim_Neighbors, and then use this illegitimate node to communicate with the other Victim_Neighbor. Eavesdropping and disguise could seriously jeopardize the network security.

3.5 Key Reinforcement

We can fix a period to reinforce the PRIVATE_KEY of every sensor nodes. According to the period each sensor node produces a random number R by itself. Sensor node uses

R and its old PRIVATE_KEY to compute a new PRIVATE_KEY through XOR operation. We can assume that the ID of this sensor node is S1. Then this sensor node S1 sends this random number R to each nearest neighbor, of course, still uses old PRIVATE-KEY to encrypt this message. Old PRIVATE_KEY couldn't be deleted till all nearest neighbors replace the old PRIVATE_KEY by the new one. But if this old PRIVATE_KEY is the original PRIVATE_KEY of this node, it will be backuped in order to provide the original PRIVATE_KEY information when a new node is to be attached in the future. When a nearest neighbor received this random number R, it computes the new PRIVATE_KEY in the same way. After that, this neighbor uses the new key to send a feedback-message. If the sensor node S1 receives this feedback-message, and successfully decrypts it by preferential using new PRIVATE_KEY, it will tell that neighbor to delete the old PRIVATE_KEY. If not, it will send that random number R which was still encrypted by old PRIVATE_KEY again to this neighbor. If it has sent the random number R to this neighbor too many times (for example, more than 4 times), it will give up and regard this neighbor as a captured node. It (S1) will delete the PRIVATE_KEY of this neighbor, and also tell other neighbors to delete that unsafe key (It only sends the ID of that captured node). The other neighbors will follow its suggestion and regard the sender S1 as a Victim_Neighbor. If one of neighbors of the sender S1, we supposes it is the sensor node S2, find that itself indeed has the PRIVATE_KEY of that captured node in its memory, then this unfortunate sensor node S2 also becomes a Victim_Neighbor. It will give the sensor node S1 a special feedback-message. How to handle this special message? We'll discuss later. After received all the feedback-messages from each nearest neighbor or knew which neighbor(s) had been captured, this sensor node S1 would delete the old PRIVATE_KEY from its memory.

As each sensor node has a few nearest neighbors, the cost of key-reinforcement is not so expensive. Sensor node can also send key-reinforcement information incidentally while sending data. How frequently the key being reinforced can determine the extent of network security. There is a trade-off between the security and the cost.

3.6 Benefit of Key-Reinforcement

3.6.1 Discovering a Captured Node

As previously mentioned, if the sensor node couldn't get feedback from any nearest neighbor in time, it'll regard that neighbor as a captured node. So each sensor node will discover whether any one of its nearest neighbors couldn't properly work any more. In addition, this sensor node itself could know whether it has already been a Victim_Neighbor. This approach could handle the following situations: the sensor node is damaged, exhausted or moved out of the communication range. In short, this sensor node couldn't respond to key-reinforcement.

If two Victim_Neighbors can communicate with each other, we must prevent them from being disguised, and deter the adversary to get messages between these two Victim_Neighbors by eavesdropping. Our corresponding countermeasure is: the Victim_Neighbor who received a special feedback-message will be deleted from the network. This unfortunate sensor node will tell its neighbors to delete itself.

Since (1) the PRIVATE_KEY of the captured node has been deleted, (2) the PRIVATE_KEY of its neighbor has been reinforced, the original captured node and its duplicate can no longer work when they come back again.

3.6.2 Increasing Sequence Number

There is one exception: In the case of not being discovered, an illegitimate sensor disguises to be a legitimate one, and communicates with the other legitimate sensor. Hence we define an increasing sequence number in the message header. This number is private for the pair of sensors which are communicating, and it should be increased one by one when each message is sent between this pair of sensors. Although the receiver couldn't distinguish the bogus message sent by the illegitimate node at once, after received the real message from the legitimate one and compared the increasing sequence number, the receiver will still discover some one must be fake in spite of that the sequence number itself is increased according to the requirement. Then both of them (the legitimate node and the illegitimate one) will be deleted. In order to decrease the wastage, when the receiver got a message, it first decrypts the message header (which contains the increasing sequence number). If the message header is correct, the receiver will continue to decrypt the rest.

3.6.3 Benefit

This approach also resists the replay attacks and DOS (denial of service) attacks to some extent, since decrypting the message header only spends a little cost. After decrypted the message header, if the increasing sequence number is incorrect, such as outdated sequence number or sequence number for the future, the receiver will discard this message.

If the reinforcement of the key takes place in time, the receiver even needn't inspect the increasing sequence number, since using new key the receiver couldn't decrypt the outdated message header or the incorrect one.

4 Optional Encryption and Decryption Algorithm

Obviously, the complexity of our key management mechanism is largely determined by the encryption and decryption algorithm(s) we use.

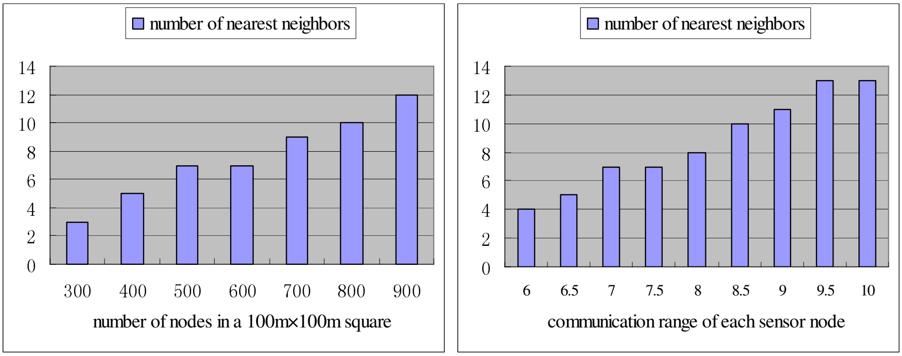
RC5

The RC5 encryption algorithm is probably a good choice in terms of energy consumption. It's a fast symmetric block cipher, and it can be implemented by hardware or software. Its word size, the number of rounds and secret key length are all variable. The energy consumption of RC5 encryption is not sensitive to the key size. In [9], authors indicate that using a 128-bit key RC5 algorithm to encrypt a 100 bytes packet may consume no more than 2μJ on average.

When each sensor node broadcasts and discovers PRIVATE-KEY, it must consume extra energy in the encryption and decryption of PRIVATE_KEY. However, this encryption as well as decryption incurs only one-time overhead cost during the whole lifetime of each sensor node, so this kind of action will not significantly increase the energy consumption.

5 Simulations

We suppose that in a square area whose length of side is 100 meters, with 500 sensor nodes randomly scattered in it(its density is 5%). We randomly scatter these 500 sensor nodes in this area for 100 times, and intend to get an average result that how many nearest neighbors each sensor node has. If the communication range of each sensor node is 7 meters, then there will be probably 7 nearest neighbors around each sensor node. That means: each sensor node only need to receive and store probably 7 PRIVATE_KEYs of its nearest neighbors. Fig.3.a. shows the differences when the number of nodes in this square changed.



(a) (b)

Fig. 3. (a) The number of nearest neighbors when the density of the 100m×100m square network is changed. The communication range of each sensor node is fixed to 7 meters. (b) The number of nearest neighbors in a 500-sensor-nodes 100m×100m square network when the communication range of each sensor node is changed.

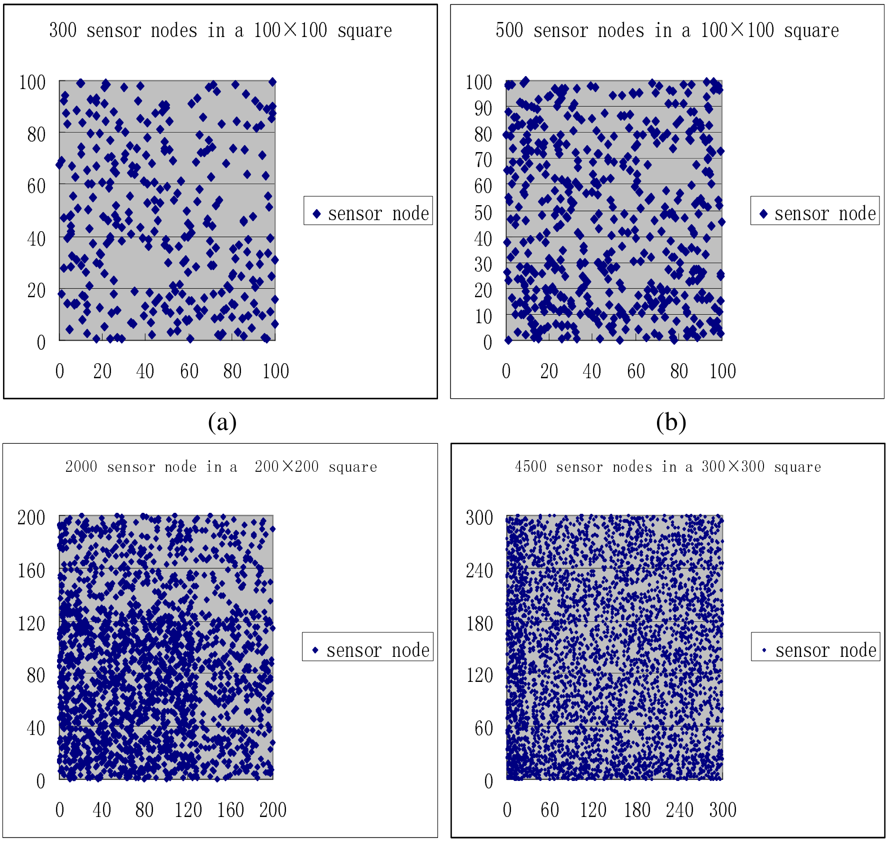
In this 500-sensor-nodes 100m×100m square network, if we only change the communication range of each sensor node, when the range is 6 meters, then there will be probably 4 nearest neighbors around each sensor node. Fig.3.b. shows these differences when the communication range of each sensor node is changed.

If we randomly scatter 2000 sensor nodes in a square area whose length of side is 200 meters(its density is 5%), compute for 100 times to get average result, when the communication range of each sensor node is still 7 meters, we'll probably get a number 7 (each node has 7 nearest neighbors) as a average result.

If 4500 sensor nodes are randomly scattered in a larger square area whose length of side is 300 meters(its density is still 5%), we run the simulation again, the average result will probably be 7.

Fig.4. shows these simulations results.

Obviously, despite the increase in the scale of the network, if its density and communication range of each sensor is fixed, each sensor will have almost the same number of neighbors. It means with limited density and limited communication range, the keys each sensor needs is limited, which is usually a tiny component of the key pool. And at the same time our mechanism provides an almost perfect connectivity.



(a) (b)

(c) (d)

Fig. 4. The position of random scattered sensor nodes in one simulation: (a)300 sensor nodes in a 100m×100m square; (b) 500 sensor nodes in a 100m×100m square; (c) 2000 sensor nodes in a 200m×200m square; (d) 4500 sensor nodes in a 300m×300m square

On the contrary, in random key predistribution mechanism, each sensor node probably needs to store dozens or even hundreds of keys for a predetermined connectivity. Furthermore, this mechanism is restricted by the scale of the network. Compared with random key predistribution mechanism, our approach remarkably reduces the memory space which is used to store each PRIVATE-KEY. Generally speaking, our mechanism is suitable for the network of any scale without requiring each sensor node to have a large memory space.

6 Conclusions

We presented a lightweight of pair-wise key management scheme for distribution wireless sensor network. We believe that if we carefully choose the encryption and decryption algorithm(s) according to the requirements, we could balance the degree of security and the requirements of simplicity / resource efficiency of resource-limited sensor nodes.

We could securely distribute pair-wise keys throughout the network. Although the encryption and decryption of keys would indeed increase energy consumption during the key sharing and discovering phase, these operations incurs only one-time overhead cost during the whole lifetime of each sensor node, so our approach remains energy effective. With limited nodes density and limited communication range, the keys each sensor needs is limited, which is usually a tiny component of the key pool. After that all the sensor nodes can communicate with each other securely. It's feasible to add a new sensor node in the network or delete an unsafe one if we need. Pair-wise keys could be reinforced to improve the security of the network. In virtue of key-reinforcement we could distinguish a captured node from other legitimate nodes, this information is very useful to damage control.

Acknowledgements. This research is supported by National Key Basic Research Program of China "973 Project" (2007CB307104 of 2007CB307100).

References

1. Camtepe, S.A., Yener, B.: Key distribution mechanisms for wireless sensor networks: A Survey. Technical Report, TR-05-07, Rensselaer Polytechnic Institute (2005)
2. Malan, D.J., Welsh, M., Smith, M.D.: A public-key infrastructure for key distribution in TinyOS based on elliptic curve cryptography. In: Proc. of the 1st IEEE Int'l Conf. on Sensor and Ad Hoc Communications and Networks, pp. 71–80. IEEE Press, Los Alamitos (2004)
3. Gaubatz, G., Kaps, J., Sunar, B.: Public keys cryptography in sensor networks—Revisited. In: ESAS. Proc. of the 1st European Workshop on Security in Ad-Hoc and Sensor Networks, pp. 2–18. ACM Press, New York (2004)
4. Huang, Q., Cukier, J., Kobayashi, H., Liu, B., Zhang, J.: Fast authenticated key establishment protocols for self-organizing sensor networks. In: Proc. of the 2nd ACM Int'l Conf. on Wireless Sensor Networks and Applications, pp. 141–150. ACM Press, New York (2003)
5. Eschenauer, L., Gligor, V.D.: A key-management scheme for distributed sensor networks. In: Proc. of the 9th ACM Conf. on Computer and Communication security, pp. 41–47. ACM Press, Washington (2002)
6. Chan, H., Perrig, A., Song, D.: Random key predistribution schemes for sensor networks. In: SP 2003. Proc. of the 2003 IEEE Symp. on Security and Privacy, Berkeley, pp. 193–213 (2003)
7. Liu, D., Ning, P.: Location-Based pairwise key establishments for static sensor networks. In: Proc. of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks, pp. 72–82. ACM Press, New York (2003)
8. Lai, B., Kim, S., Verbauwhede, I.: Scalable session key construction protocol for wireless sensor networks. In: IEEE Workshop on Large Scale RealTime and Embedded Systems (LARTES) (2002)
9. Prasithsangaree, P., Krishnamurthy, P.: On a framework for energy-efficient security protocols in wireless networks (2004)

A Stream-Data Oriented Secure Routing Protocol in Wireless Sensor Networks

Zhengjian Zhu, Qingping Tan, and Peidong Zhu

School of Computer Science, National University of Defense Technology
Changsha, Hunan, China 410073
nowaterfire@yahoo.com, 13973123266@hnmcc.com, zpd136@sina.com

Abstract. As the WSN multi-media applications go deep into the military, monitor and other data-sensitive areas, stream data have become the main data processing objects instead of scalar data in WMSN. Because of the difference between the application environments and data features of stream data and scalar data, traditional secure routing for scalar data is not fit for stream data. In this paper, SOAR, a secure route for the false data injection attack model is presented. SOAR works in the stream data transfer mode and randomly detects the false data injection attacks. SOAR guarantees that the base stations receive small percentage of false packages with rather low load.

1 Introduction

The research of wireless sensor network (WSN) currently focuses on the gathering, transmitting and processing of simple environment data [1]. However, as the monitoring environment and objects are becoming more and more complex, simple scalar data can’t satisfy the requirements of applications. It is urgent to introduce information-abundant media data, such as image, audio and video, to the WSN based environment monitoring, so that we can perform fine-grained and precise monitoring [2]. Therefore stream data have become the main data processing objects of wireless multimedia sensor network (WMSN) instead of scalar data.

Secure routing is the important security guarantee of the WMSN applications. For stream data and scalar data, their network architecture and sensor capability are very different. Furthermore, their data content, size and correlation are greatly different. So the traditional secure routing protocols for scalar data in WSN cannot be applied to the stream data in WMSN.

This paper proposes a stream-data oriented secure routing protocol (SOAR) for WMSN. SOAR is applicable to the real-time and energy-constrained applications.

The rest of this paper is organized as follows. Section 2 discusses related work. Section 3 presents SOAR in detail. Section 4 gives complete security analysis. Finally, Section 5 gives concluding remarks and directions in future work.

2 Related Work

Routing and energy efficiency are the main focus of the general routing protocols, however, the security issue is not considered [3], so the general routing protocols are

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 845–853, 2007.

vulnerable to various attacks. For example, GEAR [4] needs exchanging location information, so a malicious node can put itself on the routing path by broadcasting fake location information. If the malicious node participates in data transmission, it can perform many attacks such as the selective forwarding attack.

The multi-path routing mechanism proposed by Ganesan etc. [5] can resist the selective forwarding attack. The main idea is to build multiple paths from the source to the destination, and transmit the copies of one packet on multiple paths. So long as there is no malicious node performing selective forwarding on one single path, the packet can be delivered to the destination.

INSENS proposed by Deng [6] limits the destruction of a malicious node to some extent. And it can continue providing routing service while not excluding the malicious node.

SEF proposed by Ye [7] focuses on identifying false data. When an event happens, SEF elects one center node CoS from all the sensors which detect the event. CoS gathers all the detecting data and generate an integrated report. Then CoS broadcasts the report to all the sensors which detect the event. If the detecting node agrees with the report, then it generates a message authentication code (MAC) and transmits it to CoS. After CoS gathers enough MACs, it notifies the sink of the event. However, the event reports about which CoS haven't gathered enough MACs will be abandoned.

Zhu etc. proposed an interleaved hop-by-hop authentication (IHOP) scheme in [8]. IHOP guarantees that the base station will detect any injected false data packets when no more than a certain number t of nodes are compromised.

if the existing research is applied to stream data in WMSN, the following problems will appear:

- The energy consuming is too large

The existing work mainly focuses on protecting every data packet. However the stream data amount is very huge. If every packet is processed securely, the energy consuming will be definitely very large. Obvious it is not practical for energy constrained sensors in WMSN.

- The additional delay worsens QoS

When the secure mechanisms such as SEF are validating data, in order to carry out voting they need transmit each packet repeatedly between event detecting nodes and data gathering node. Obviously this will lead to large delay between neighboring packets. However the stream data have temporal and content correlation, so the delay will greatly worsen the application QoS.

EASY proposed by us [9] adopts random detection mechanism to reduce the packets which need be validated. Then the whole network load is decreased. But there is a strict limit to EASY. It's that the routers and cluster heads can't be captured.

We propose SOAR in the paper. SOAR can get the similarly same security level of EASY without the limit.

3 Soar Protocol

The goal of SOAR is to try to avoid the false data injection attack, and at the same time increases little computation and low communication overload. It assures that the percentage of the false packets in the total packets received by the sink is approximately 0. And the whole data stream has small delay and low consuming energy.

3.1 Assumptions and Main Idea

The main idea of SOAR is as follows: SOAR randomly samples and validates some packets of the data stream, so that the amount of the packets to be validated is largely reduced. Therefore, the additional network overload induced by security is finally decreased.

Before we describe SOAR in detail, we firstly give the following assumptions about the application background.

1.The nodes are deployed densely.

2.The entire field is virtually partitioned into non-overlapping hexagonal cells of equal areas. Every node in a cell can directly communicate with another node in a neighbor cell.

3.The nodes know the location of sink and themselves. They use geography route.

4.There exists authenticity validation mechanism to monitor the same event among the multiple nodes.

3.2 SOAR Description

The SOAR protocol has three phases: node initialization, security initialization, and data processing phase.

Node Initialization Phase

During the SOAR node initialization phase, the security information is distributed to all sensor nodes. The main process is as follows:

1.The identifier U_i and the master secret key K are loaded into the nodes. U_i is unique in the network and K is used to derive some necessary keys in security initialization phase.

2.Sink load the necessary security arithmetic into the nodes.

3.We generate ID-based identity key KS_i shared with the sink for each sensor.

$$KS_i := H\left(K \| U_i\right)$$

4.The entire field is virtually partitioned into non-overlapping hexagonal cells of equal areas [10]. If the communication radius of every sensor node is R, we design hexagonal cells with the maximal lateral dimension R/2. With this choice, every node in a cell can directly communicate with another node in a neighbor cell.

Security Initialization Phase

After node initialization phase, all sensor nodes are uniformly randomly deployed to the monitoring area. Then every sensor node Uperforms the following:

4.It obtains the location $(x_c{}^{‹}\ y_c)$ of the center of the cell in which it is residing on using a localization scheme [11].

5.It obtains the location $(x_n{}^{‹}y_n)$ of the center of the neighbor cell.

6.It uses the master key K to derive the necessary keys.

Neighbor node key $KN_{u1,u2} := H\left(K \| u1 \| u2\right)$

Neighbor cell key $KC_{c1,c2} := H\left(K \| x1 \| x2 \| y1 \| y2\right)$

The (x1, y1) and (x2, y2) are the location of cell c1 and cell c2.

7. It deletes the master key K.

8. One transmission node symbolized as TR will be randomly selected in every cell. The random selection process will be done every T1. T1 is a time parameter. By the periodic random selection, we can avoid that TR captured by enemy is always a TR. The other node that is not TR will be the scout node symbolized as SC. The SC will scout the behavior of the TR in the same cell.

Similar to [12], we assume that the initialization phase after deployment is secure, i.e., none of the nodes is captured.

Data Processing Phase

We make an assumption that from the source node to sink, the cell path is C_1, C_2…,C_n and the TR path is TR_1,TR_2,…TR_n. TR_1 is source node and TR_n is sink. In the phase, source node, TR and SC respectively finish the following work.

- **Source node**

1. It adds U_i and the variable SOU into the packet.
2. It assigns 1 to the SOU .
3. It encrypts the packets with its neighbor cell key $KC_{c1,c2}$.
4. It calculates the MAC with the identity key KS_{TR1}. We symbolize the MAC as MACKS.
5. It adds the MACKS into the packet.
6. Finally it sends the packet to the next cell.

- **TR_i**

1. TR_i accepts the packets that are sent to the cell where it locates.
2. TR_i waits for time T2 to examine whether the demurral packets arrive.
3. If no demurral packets arrive, it shows that no malicious nodes tamper the packets in the transmission from the TR_{i-1} to TR_i.

If $SOU=1$, then TR_{i-1} is the source node. The accepter is TR_2. TR_2 will randomly sample the packets with probability P. For the packet sampled by TR_2, the packet will be sent back to C1. For the node in C_1, the node that thinks the packet true will add two MACs into the packet. One MAC is calculated with neighbor node key. The other MAC is calculated with its identity key. Then the node sends the packet to TR_2. If TR_2 doesn't get enough MACs, it will drop the packet. If TR_2 gets enough MACs, it will assign 0 to the SOU and encrypt the packets with its neighbor cell key $KC_{c2,c3}$. Finally it sends the packet to C_3. For the packet that isn't sampled, TR_2 assigns 0 to the SOU. Then it encrypt the packets with its neighbor cell key $KC_{c2,c3}$ and sends the packet to C_3.

If $sou = 0$, then the directly sender of the packet isn't the source node. TR_i encrypts the packets with its neighbor cell key $KC_{ci,c(i+1)}$. Then it send the packet to C_{i+1}.

4. If the demurral packets arrive, it shows that malicious nodes may tamper the packets in the transmission from the TR_{i-1} to TR_i.

TR_i sends the suspicious packet back to C_{i-1}. If the packet have the MACs of the most node in the C_{i-1}, TR_i will send the packet to C_{i+1}. Otherwise, it will drop the packet and begin the malicious node excluding process [13].

- **SC**

SC will scout the behavior of the TR in the same cell. If the packet that the TR sends out is inconsistent with the packet that the TR accepts, SC will send the demurral packet to the next cell.

The voting in the source cell is for validating the content of the packet. For example, whether is the temperature 38 centigrade? The voting in the other cells is for judging whether the packets are tampered in the transmission process.

4 Security Analysis

In this section we first analyze the miss ratio of SOAR by simulation. Then we compare SOAR with SEF and EASY.

4.1 Analysis for Miss Ratio

Threat Model: There are D nodes in the network. The entire field is virtually partitioned into non-overlapping hexagonal cells of equal areas. The enemy randomly captures the nodes. The percentage of the nodes captured by the enemy in the total nodes is α . The nodes captured by the enemy will become the malicious nodes that have all security information in the original nodes. The malicious nodes will tamper the packets which pass through themselves. The probability of which the TR_2 validates the packet authenticity is β. In the voting examination, the percentage of the nodes which agree with the packet in the total cell nodes must be greater than or equal to ω. The number of the packets of the false data stream is N. During the time for transmitting data, the TR has changed for M times in every cell. For simplicity, we suppose that the time for which the network excludes the malicious nodes can be ignored.

Definition 1: *Miss ratio.* The malicious nodes continuously send the N false packets to sink. When the packets transmission ends, we define the miss rate μ as the expectation of the percentage of the false packets received by the sink in the total false packets.

We classify the false packet as unreal packet and distortion packet

Definition 2: *Unreal packet.* The content of the packet differs from the fact. And no nodes tamper the packet except the source node.

Definition 3: *distortion packet.* The packet is tampered by the other nodes besides the source node.

In SOAR, the TR_2 takes full responsibility for identifying the unreal packet. The SC takes full responsibility for identifying the distortion packet.

When $\alpha < \omega$, the unreal packet sampled by the TR_2 can be identified as long as the TR_2 isn't captured. If the TR_2 is tampered, it will purposely let the unreal packet pass. We periodically and randomly choose the TR_2, so it can largely alleviate the effect of the capture of the TR_2. In the transmission, the SCs can identify the distortion packet due to $\alpha < \omega$.

When $\alpha > \omega$, the false packets can't be identified.

We use simulation to investigate the relation between μ and various parameters,

For all simulation, we assign $\left\lceil \frac{N}{100} \right\rceil$ to M.

Figure 1 shows the relation between μ and α, β, N. We get the following conclusions.

- If β and N are fixed, μ is always increasing along with the increasing of α. When $\alpha > \omega$, μ will quickly get the maximum 1. The reason is that with the increasing of α, the probability of which the TR_2 is malicious node will be higher. When $\alpha > \omega$, SOAR can't identify the false packets. All false packets can arrive at the sink.
- If α and N are fixed, μ will decrease along with the increasing of β. When $\beta = 1$, μ doesn't decrease to zero. This is because the time when TR_2 finds the false packets also is also advanced with the increasing of β. Because TR_2 may be malicious node, when $\beta = 1$, μ will not decrease to zero.
- If α and β are fixed, μ will decrease along with the increasing of N. The limit of μ is zero. This is because the percentage of false packets will be lower with the increasing of N.

4.2 The Compare Among the Protocols

In order to compare SOAR with SEF and EASY, We firstly give the following definition.

Definition 4: *resist ratio*. We define the resist ratio θ as the percentage of the false packets that are identified and drop in the total false packets sent by the malicious node.

Because SEF won't exclude malicious nodes, we can't directly compare SEF with SOAR and EASY. In the simulation, we join the malicious nodes excluding phase with SEF. After excluding the malicious nodes, SEF will identify and drop the false packets that are sent by it.

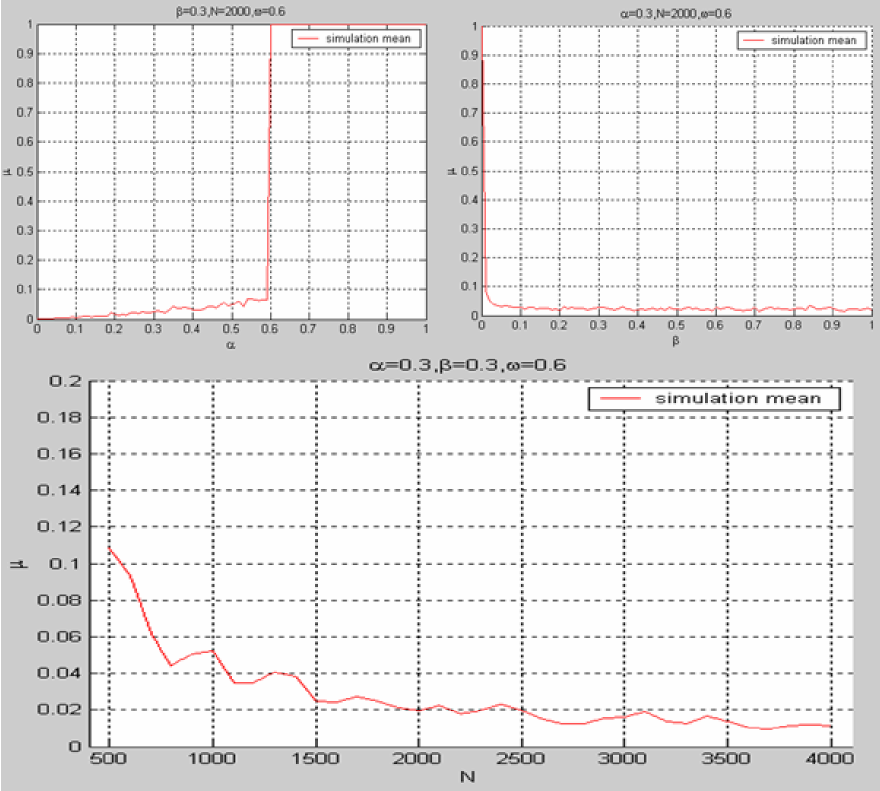


Fig. 1. The relation between μ and the various parameters

By simulation, we compare SOAR with SEF and EASY. Figure 2 shows the relation between θ and α, β, N. We get the following conclusions.

- When $\alpha > \omega$, none of three protocols can identify the false packets.
- When $0 < \alpha < \omega$, SEF has the highest value of θ, and SOAR has the lowest value of θ. But the difference in the resist ratio of the three protocols is very small. The difference is no more than 10%.

SEF can get the highest value of resist ratio, which is based on that SEF examines every packet in every hop. The security load of SEF is largely greater than that of SOAR and EASY. In order to work effectively, EASY requires that the routers and the cluster heads can't be captured, but most applications can't satisfy the condition. So SOAR can work as well as SEF with lower load in the most applications.

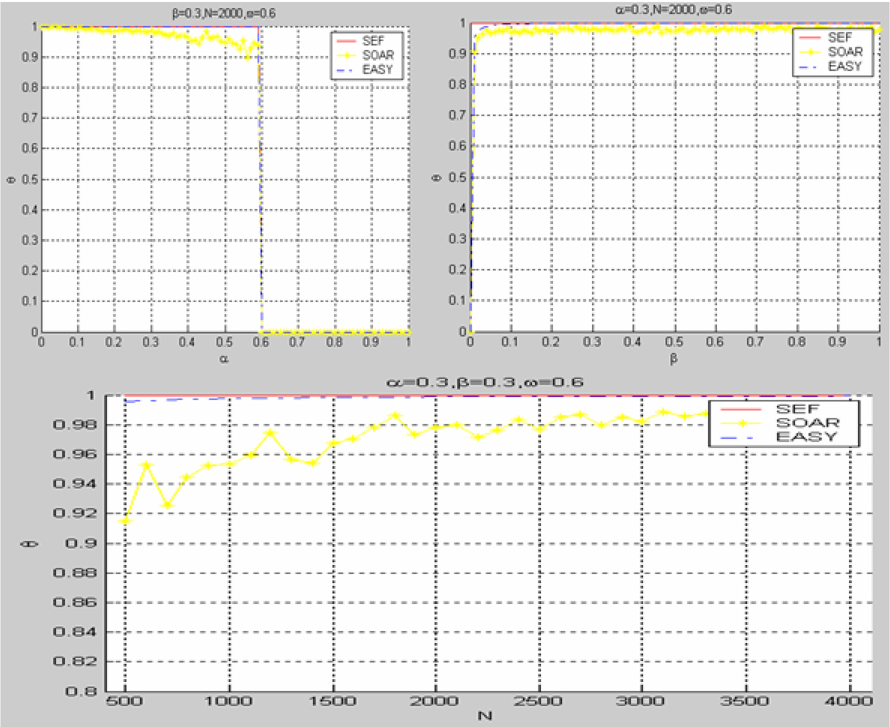


Fig. 2. The comparison among SOAR, EASY and SEF in θ

5 Conclusion

By reducing the number of the packets to be validated, SOAR lowers the load which security protection generates. By the simulation, we prove that in most cases SOAR can work as well as SEF with lower load. And SOAR can eliminate the limit of the assumption of EASY. Aiming at the attack of false packet injection, SOAR can effectively resolve the contradiction between security requirement and additional load in stream-data applications.

References

1. Akyildiz, F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: A survey on sensor networks. IEEE Communications Magazine 40(8), 102–114 (2002)
2. Holman, R., Stanley, J., Ozkan-Haller, T.: Applying Video Sensor Networks to Nearshore Environment Monitoring. IEEE Pervasive Computing 2(4), 14–21 (2003)
3. Estrin, D., et al.: Directed Diffusion for Wireless Sensor Networking. IEEE/ACM Transactions on Networking (Februray 2003)

4. Yu, Y., Govindan, R., Estrin, D.: Geographical and Energy Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks. UCLA Computer Science Department, Tech. Rep. UCLA/CSD-TR-01-0023 (May 2001)
5. Ganesan, D., Govindan, R., Shenker, S., Estrin, D.: Highly-resilient,energy-efficient multipath routing in wireless sensor networks. Mobile Computing and Communications Review (MC2R) 1(2) (2002)
6. Deng, J., Han, R., Mishra, S.: INSENS: intrusion-tolerant routing in wireless sensor networks. Technical Report CUCS-939-02, Department of Computer Science, University of Colorado (2002)
7. Ye, F., et al.: Statistical En-Route Filtering of Injected False Datasensor Networks. In: Proc. IEEE INFOCOM, Hong Kong (2004)
8. Zhu, S., et al.: An Interleaved Hop-by-Hop Authentication Scheme for Filtering of Injected False Data in Sensor Networks. In: Proc. IEEE Symp. Security and Privacy, Oakland, CA, pp. 259–271 (May 2004)
9. Zhu, et al.: An effective secure routing for false data injection attack in wireless sensor network. In: Proceedings of the Asia-Pacific Network Operations and Management Symposium, pp. 457–465 (2007)
10. Stuber, G.L.: Principles of Mobile Communication, 2nd edn. Kluwer Academic, Boston (2001)
11. Lazos, L., et al.: ROPE: Robust position estimation in wireless sensor network. In: IPSN 2005. Proc. Int. Symp. Inform. Process. Sensor Networks, pp. 324–331. IEEE, CA (2005)
12. Anderson, R., Chan, H., Perrig, A.: Key infection: Smart trust for smart dust. In: ICNP 2004. Proc. IEEE Int. Conf. Network Protocols, pp. 206–215. IEEE Comput. Soc., CA (2004)
13. Wang, G., et al.: On supporting Distributed Collaboration Sensor Networks. In: Proc. MILCOM (2003)

An Improved Key Management Scheme for Heterogeneity Wireless Sensor Networks

Chunguang Ma, Zhiguo Shang, Huiqiang Wang, and Guining Geng

College of Computer Science and Technology, Harbin Egineering University
150001 Harbin, China
{machunguang,wanghuiqiang}@hrbeu.edu.cn,
{szg202,gengguining}@163.com

Abstract. The growing popularity of wireless sensor networks has brought increasing attention to many security issues for such networks. In these security issues, key management is one of the most challenging and in dire need of solving problems. A lot of research has been mainly concentrated on key management for homogeneous wireless sensor networks. However, such homogeneous networks could restrict them to have a long network lifetime as well as to improve network connectivity. Recent research has shown that heterogeneous wireless sensor networks have greater performance and reliability. In this paper, we propose a key management scheme for heterogeneous wireless sensor networks to improve the random key pre-distribution scheme using deployment knowledge of nodes and the prior area deployment information. The performance evaluation and security analysis show that our scheme can substantially improve the network connectivity with low complexity and significant reduction on storage requirement, and enhance the network resilience against node capture, compared with existing key management schemes.

Keywords: Heterogeneous wireless sensor networks; Key management; Key pre-distribution; Security.

1 Introduction

With the recent technology improvement of micro-electro-mechanical, wireless communication, and information networks, wireless senor networks are widely applied in various applications such as object tracking, environment monitoring and many military applications in the future. Wireless senor networks consist of a large number of sensor nodes which have limited energy resources, computation ability and wireless communication range. Generally, wireless sensor networks are deployed in hostile environments and operated in unattended mode. Hence, security mechanisms that provide confidentiality and authentication are critical for the operations of many sensor applications. However, due to resource constraints of sensor nodes, it is infeasible to use public key cryptography and key distributed [1]. Instead, sensor networks should use symmetric cryptography technology, low-power authentication mechanism and Hash function. Now, a generally accepted method is the key

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 854–865, 2007.

pre-distribution model, it needs to setup keys or key materials in senor nodes before the networks are deployed.

At present, there are many key management schemes using the key pre-distribution model directly. The simplest key management scheme is a network wide shared key. Unfortunately, the compromise of even a single node in network would reveal secret key and thus allow decryption of all network communications. Another simplest scheme is to pre-configure the network with a shared unique key between each pair of nodes. Though it achieve the strongest resilience toward node capture, it is not suitable for large-scale wireless sensor networks. For improving security and reducing memory, Eschenauer and Gligor [2] first presented a basic random key pre-distribution scheme. In order to improve the network resilience against node capture attacks, Chan et al. [3] proposed a "*q-composite*" scheme based on the basic random key pre-distribution scheme. In their scheme, every two nodes need to share at least q ($q \geqslant 2$) common keys to establish a secure link between them. For reducing the size of key ring, Ren et al. [4] made an improvement on structuring and distributing key ring based on Eschenauer et al.' works. Chan et al. [3] proposed a random-pairwise keys scheme according to Erdös and Rényi theorem. Du et al. [5] presented a matrix-computation-based key pre-distribution scheme. The scheme exhibits a threshold property that when the number of compromised nodes is less than the threshold, the probability that any other nodes are affected is close to zero. Liu et al. [6] proposed a key management scheme based on polynomial in which they use a t-degree bivariate polynomial pool instead of key pool. In order to use keys in the key ring more effectively, Donggang Liu and Peng Ning [7] introduced the position information into the scheme of random-pairwise keys scheme. Du et al. [8] proposed a key pre-distribution scheme using prior deployment knowledge. Fang Liu et al. [9] replaced keys in memory with the system configuration information which can produce keys or key materials in the key pre-distribution phase.

All key management schemes mentioned previously are mainly considered homogeneous wireless sensor networks. However, such networks have a poor performance and scalability [10]. Research [11~14] indicates that heterogeneity wireless sensor networks (HWSNs), which consist of different types sensor nodes, can significantly improve networks performance. At present, there are only a few key management schemes for HWSNs. For instance, Kejie Lu et al. [15] and Xiaojiang Du et al. [10] proposed a key management scheme for HWSNs based on the random key pre-distribution scheme and polynomial key pre-distribution scheme separately, but the improvement of security and connectivity, the reduction of required memory in such schemes are not too ideal.

In this paper, we present an improved key management scheme for HWSNs based on the random key pre-distribution scheme. The scheme exploits deployment knowledge of nodes and the prior region deployment information. In the scheme, we divide the network sensing region into several sub regions; divide the key pool into several key pools, and divide nodes into several groups. Nodes in a certain group randomly select some keys from the corresponding key pools are deployed at the corresponding sub regions. This can improve the probability that neighboring nodes share common keys at a certain extent.

The remainder of this paper is organized as follows. We fist introduce the model of networks in section 2. In section 3, we describe key management in detail. Section 4 is the performance evaluation and section 5 is the security analysis. After analyzing the experimental results, we conclude the paper with section 6.

2 Network Model

In the paper, we consider a simple HWSNs consisting of two types of sensor nodes: a small number of powerful cluster head nodes (H-sensors) and a large number of data sensing nodes (D-sensors). H-sensors have sufficient energy resources, great computation ability and wild communication range. D-sensors have limited computation ability, energy resources and communication range. D-sensors are used to sense data in the detecting region. H-sensors provide data collecting, fusion and transport. An example of the network model is shown in Fig 1, where suppose the number of network deployment sub regions is 9. We list assumptions of HWSNs below.

(1) H-sensors are equipped with tamper-resistant hardware. It is reasonable to assume that a H-sensor can not expose all the information on stored it when it was compromised.

(2) Assume that if an adversary compromises a D-sensor, he can extract all the key material, data, and code on stored it.

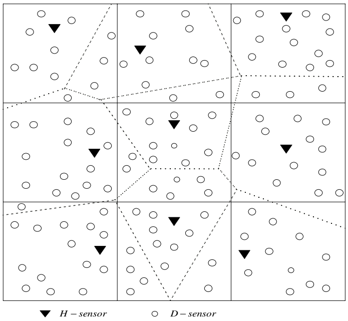


Fig. 1. The model of HWSNs

3 The Key Management Scheme for HWSNs

In this section, we present a key management scheme specifically designed for heterogeneous wireless sensor networks as well as setting up the key pools $S_{i,\,j}$ and determining $|S_G|$.

3.1 The Scheme Description

Our scheme consists of three phases: key pre-distribution, shared-key discovery, and path-key establishment. Because we adopt the pre-deployment knowledge, prior area information and some powerful nodes, all phases are considerably different from the random key pre-distribution scheme [1].

Step 1: key pre-distribution phase.
This phase is performed offline and before the deployment of sensor nodes. First we need to divide the network sensing area D into $u \times v$ sub regions $D_{i,j}$, $1 \le i \le u,$ $1 \le j \le v,$

with $D_{i,j}$ corresponding to the key pools $S_{i,j}$ and deployment group $G_{i,j}$. After completion of key pools setup, for each D-sensor and H-sensor in $G_{i,j}$, we randomly select m and M keys from its corresponding key pool $S_{i,j}$, and load these keys and its identifiers into the memory of the node.

Step 2: shared-key discovery phase.
After deployment, HWSNs are divided into multiple clusters according to the clustering algorithm [14]. After completion of clusters forming, each node needs to discover whether it shares any keys with its neighbors. To do this, each D-sensor could broadcast message: α, $E_{k_i}(\alpha, ID)$, $1 \le i \le m$, where α is a random number. The decryption of $E_{k_i}(\alpha, ID)$ with the proper key by a recipient would reveal α, ID and establish a shared key with the broadcasting node.

After the above step, most D-sensor and H-sensor can establish direct shared-key. These D-sensors are referred to as several 1 hop neighbor nodes of H-sensors, which is defined as follows:

Definition 1. A D-sensor that shares at least one key with H-sensor and its communication range covers the H-sensor is referred to as a 1-hop neighbor node of the H-sensor [10].

Step 3: the path-key establishment
After shared-key discovery phase, there is a direct shared-key graph in the cluster and between clusters. This graph consists of the nodes that establish the direct shared-key and their security link. The rest of nodes that can not establish the direct shared-key could find the path key through this graph.

After this phase, D-sensors that haven't established the direct shared-key with H-sensor need to find (k+1)-hops path-key (for k=1, 2,…). These D-sensors are referred to as several (k+1)-hops neighbor nodes of H-sensors, which is defined as follows:

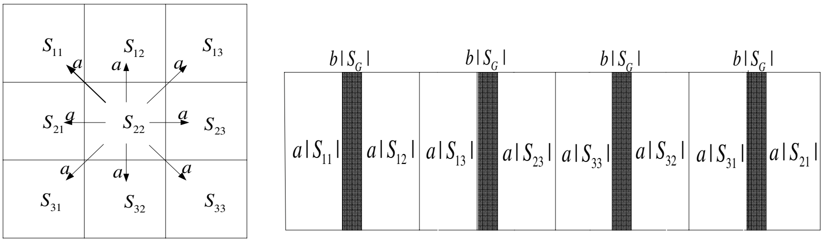
Definition 2. A D-sensor that hasn't established k-hops path-key with H-sensor, but shares at least one key with some k-hops neighbor nodes and its communication range covers the k-hops neighbor nodes is referred to as a (k+1)-hops neighbor node of the H-sensor [10].

3.2 Setting Up $S_{i,j}$

In this subsection, we show how to assign keys to each key pools $S_{i,j}$, such that $S_{i,j}$ and neighbors of $S_{i,j}$ have a certain number of common keys. Based on Du et al.'s scheme [8], we adopt a new method of setting up key pools and determining $|S_G|$ in our scheme, it is different from Du et al.'s scheme. In our scheme, we have:

(1) Two horizontally, vertically and diagonally neighboring key pools share exactly $a|S_G|$ keys, where $1/8 \leqslant a \leqslant 1/4$. This property is illustrated in Fig.2 (a).

(2) To the group $G_{i,j}$ and its corresponding the key pool $S_{i,j}$,$a|S_{i-1,j-1}|$ and $a|S_{i-1,j}|$, $a|S_{i-1,j+1}|$ and $a|S_{i,j+1}|$, $a|S_{i+1,j+1}|$ and $a|S_{i+1,j}|$, $a|S_{i+1,j-1}|$ and $a|S_{i,j-1}|$, share exactly $b|S_G|$ keys, where $a|S_{i-1,j-1}| \subset S_{i,j}$,…, $a|S_{i,j-1}| \subset S_{i,j}$, $0 \leqslant a \leqslant 1/4$, $8a-4b = 1$. This property is illustrated in Fig.2 (b).



(a) *a* (b) *b*

Fig. 2. The relationship of overlapping facters and key pools

Given the global key pool S and the overlapping factor a and b, we now describe how we can select keys for each key pool $S_{i,j}$ for i=1,…, u and j=1,…,v. The following procedure describes how we choose keys for each key pool:

(1) For group $S_{1,1}$, select$|S_G|$ keys from S, then remove these$|S_G|$ keys from S.

(2) For group $S_{1,j}$, for j=2, 3, ... , v, select $a|S_G|$ keys from $S_{1,j-1}$, then select $\sigma = (1\text{-}a)|S_G|$ keys from S, and remove σ keys from S.

(3)For group $S_{i,j}$, for i=2, 3,…u, and j=1,2,…v, select $a|S_G|$ keys from each of $S_{i-1,j}$, $S_{i,j-1}$, $S_{i-1,j-1}$and $S_{i-1,j+1}$ if they exist; then select σ keys from S ,and remove σ keys from S. According to the difference between the deployment sub regions of nodes in each group, and the difference between neighboring key pools, σ may have the following several values:

$$\sigma = \begin{cases} (1-2a+b)\cdot|S_G|, & (\text{for } j\text{=1}) \\ (1-4a+2b)\cdot|S_G|, & (\text{for } 2\le j\le v\text{-1}) \\ (1-3a+b)\cdot|S_G|, & (\text{for } j= v) \end{cases} \tag{1}$$

Now we can calculate the size of the key pool $|S_G|$ for each group. Since keys selected from the other groups are all distinct, the sum of all the numbers of keys should be equal to$|S|$.Therefore, we have the following equation:

$$|S_G| = \frac{|S|}{uv-(4uv-3u-3v+2)a+2(uv-u-v+1)b} \tag{2}$$

4 Performance Evaluation

In this section, we evaluate the performance of our scheme against metrics which are connectivity and the probability of being a k-hops neighboring node.

4.1 Connectivity

Connectivity is the probability that two or more nodes share common key or key material [16]. In the cluster, the connectivity may consist of four types of probability: the probability that D-sensor and H-sensor share at least one key at the same or neighborring regions, the probability that two D-sensors share at least one key at the same or neighboring regions.

We calculate the connectivity $P(\alpha,\beta)$, the probability that two nodes share at least one key. $P\ (\alpha,\beta)$ can be expressed as 1- P_r[two nodes do not share any key].To calculate $P(\alpha,\beta)$, we need to calculate P_r [two nodes do not share any key] at first. We adopt the similar overlapping key pool method used in [8].The first node that must be any of the D-sensor nodes selects i keys from $\alpha|S_G|$ shared keys, it then selects the remaining m-i keys from the non-shared keys. The second node that might be D-sensor or H-sensor node selects β keys from the remaining $|S_G|$-i keys from its key pool. Therefore, $P(\alpha,\beta)$ can be calculated as follows:

$$\begin{aligned} P(\alpha,\beta) &= 1-P_r(\text{two nodes do not share any key}) \\ &= 1-\frac{\sum_{i=0}^{\min(m,\alpha|S_G|)}\binom{\alpha|S_G|}{i}\binom{(1-\alpha)|S_G|}{m-i}\binom{|S_G|-i}{\beta}}{\binom{|S_G|}{m}\binom{|S_G|}{\beta}} \end{aligned} \tag{3}$$

where α is 1or a which present nodes at the same and neighboring region separately; β is m or M. If $\alpha=1$, $\beta=m$ or $\beta=M$, $P\ (\alpha\ ,\ \beta)$ can be simplified in the following:

$$P(1,m)\ \ =1-\frac{[(|S_G|-m)!]^2}{|S_G|!(|S_G|-2m)!} \tag{4}$$

$$P(1,M)=1-\frac{(|S_G|-m)!(|S_G|-M)!}{|S_G|!(|S_G|-m-M)!} \tag{5}$$

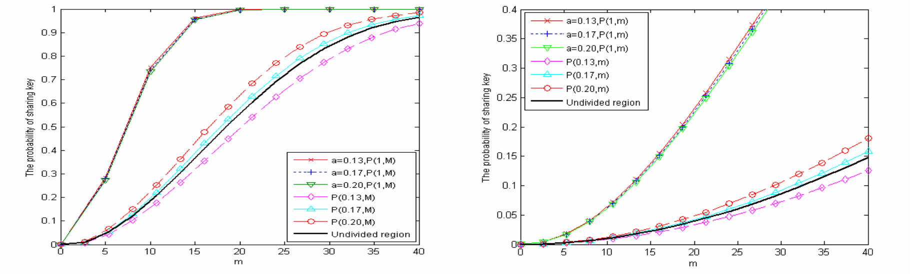


Fig. 3. Connectivity of adopting different *a*

Under the deployment area of HWSNs is divided into 9 sub regions, $|S|$=10,000, M=20m, the overlapping factors are 0.13, 0.17 and 0.20 respectively, Fig.3 illustrates the connectivity of adopting different a versus the number of keys each sensor node carries. This figure show that, with the increasing number of keys each node can carry in its memory, the probability $P(\alpha,\beta)$ shortly increase under the same overlapping factors. Moreover, $P(1, m)\geq P$, $P(1, m)\geq P(a, m)$ and $P(1, M)\geq P$, $P(1, M)\geq P(a, M)$, do not change along with the overlapping factors changing; $P(a, m)$ and $P(a, M)$ gradually increase along with overlapping factors increasing, when overlapping factor a=0.17, being $P(a, m)\geq P$ and $P(a, M)\geq P$.

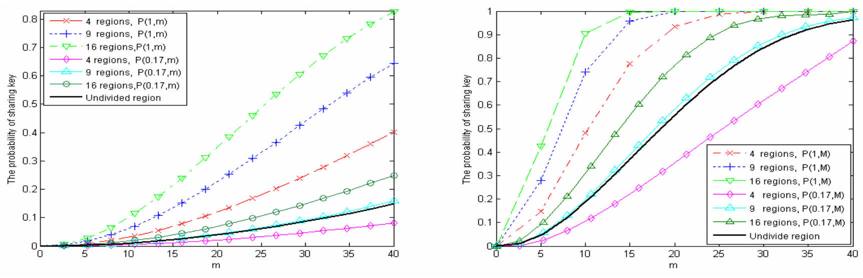


Fig. 4. Connectivity of the deployment area being divided different sub regions

Fig.4 shows the plot of the connectivity of the deployment area being divided different sub regions, $P(\alpha, \beta)$, as a function of the number of keys each sensor node carries, m, for the deployment area of HWSNs is divided into 4, 9and 16 sub regions, $|S|$=10,000, M=20m, the overlapping factors a=0.17. We can see from Fig.4, with the increasing number of the deployment sub regions, the probability $P(\alpha, \beta)$ increases, and the probability $P(1, m)$ and $P(1, M)$ are higher than the probability that the deployment region is not divided. Moreover, if the number of deployment sub regions increased to 9, the probability $P(a, m)$ and $P(a, M)$ are also higher than the probability that the deployment region is not divided. This means that with the increase of the direct shared-key probability, many nodes can establish shared-key in the shared-key discovery phase, and there is no need to find path-keys in the path-key establishment phase, thus reduce the cost of communication overhead for finding path-keys.

To achieve desired global connectivity of wireless sensor networks P_c, Eschenauer and Gligor[1] calculate the necessary expected node degree d in terms of the size of the network n as:

$$d = \frac{(n-1)}{n}[\ln(n) - \ln(-\ln(P_c))] \qquad (6)$$

In the wireless sensor networks, let the actual deployment density of nodes n', the probability that two nodes share at least one key P_{local}. To achieve desired global connectivity P_c, P_{local} is defined by: $P_{local} = \frac{d}{n' - 1}\frac{(n-1)}{n}[\ln(n) - \ln(-\ln(P_c))]$.

From Fig.3 and Fig.4, we can see clearly that we can adjust the number of the deployment sub regions and the size of overlapping factors to insure $P(a, m) \geq P_{local}$ and $P(a, M) \geq P_{local}$, and make sure nodes in clusters achieve a certain required connectivity, thus achieve the desired global network connectivity P_c.

4.2 The Probability of Being a k-Hops Neighboring Node

In cluster, because D-sensors have limited computation ability, energy resources and communication range, D-sensors need to make some of them establish 1-hop shared-key with H-sensor, and become 1-hop neighboring nodes; some establish 2-hops path-key, and become 2-hops neighboring nodes; …, and so on. As shown in Fig.5, D-sensors and H-sensors might establish two types of 1-hop shared-key and four types of 2-hops path-key, namely, D-sensors may have two methods to become 1-hop neighboring nodes, four methods to become 2-hops neighboring nodes. However, not each cluster contains all kinds of neighboring nodes, which kinds of neighboring nodes are formed according to the deployment sub regions of nodes in cluster. In Fig.5, P_1^a and P_1^1 respectively express the different probability of being 1-hop neighbor, P_2^{11} , P_2^{1a} and P_2^{aa} , P_2^{a1} respectively represent the different probability of being 2-hops neighbor.

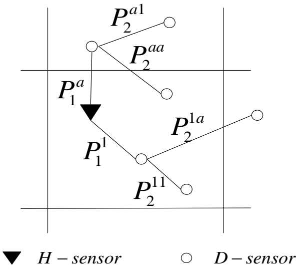


Fig. 5. The sorts of the probability of being 1/2-hop(s) neighboring nodes

According to definition 1, we can calculate the probability that any a D-sensor *x* is 1-hop neighbor node as: $P_1^1 \equiv P(1,M)$ and $P_1^a \equiv P(a,M)$.

Let *N* denote the number of D-sensors in cluster, if the probability of 1-hop shared-key is P_1^1 , thus the average number of 1-hop neighbor nodes is $\lfloor P_1^1 \times N \rfloor$. Suppose another D-sensor node *y*, *y* is not 1-hop neighbor node, let the probability that *y* share at least one key with 1-hop neighbor nodes *P*

$$\begin{aligned} P_{12} &\equiv P_r(y \text{ share at least one key with 1-hop neighbor nodes}) \\ &= 1 - P_r(y \text{ does not share any key with 1-hop neighbor nodes}) \\ &= 1 - \left(\binom{|S_G|}{m}\binom{|S_G|-m}{m} \Big/ \binom{|S_G|}{m}^2 \right)^{\lfloor P_1^1 \times N \rfloor} = 1-(1-P(1,m))^{\lfloor P_1^1 \times N \rfloor} \end{aligned} \tag{7}$$

Thus, according to definition 2, the probability of *y* being a 2-hops neighbor node is:

$$\begin{aligned} P_2^{11} &= P_r((\text{y is not 1-hop neighbor node}) \cap \\ &\qquad (\text{y share at least one key with 1-hop neighbor nodes})\,) \\ &= (1-P_1^1) \times P_{12} \\ &= (1-P(1,M)) \times [1-(1-P(1,m))^{\lfloor P_1^1 \times N \rfloor}] \end{aligned} \tag{8}$$

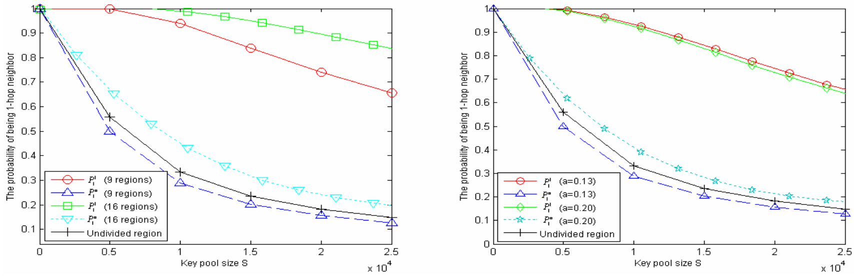
For the same reason, we can obtain other types of the probability of being 2-hop neighbor node as follows:

$$P_2^{1a} \equiv (1-P(a,M))\times[1-(1-P(a,m))^{\lfloor P_1^1\times N\rfloor}] \qquad (9)$$

$$P_2^{a1} \equiv (1-P(a,M))\times[1-(1-P(1,m))^{\lfloor P_1^a\times N\rfloor}] \qquad (10)$$

$$P_2^{aa} \equiv (1-P(1,M))\times[1-(1-P(a,m))^{\lfloor P_1^a\times N\rfloor}] \qquad (11)$$

Similarly, we can calculate the probability of a D-sensor node being a 3-hops neighbor node, 4-hops neighbor node, etc. Suppose *N*=100, *m*=20, *M*=200, Fig.6 shows how the probability of being 1-hop neighbor node varies with the size of key pool *S*. In Fig.6 (a), we can see that the probability of being 1-hop neighbor node increases with the increasing number of the deployment sub regions when the overlapping factor *a*=0.13. As shown in Fig.6 (b), let the number of the deployment sub regions is 9, the probability of a D-sensor node at neighboring sub region being 1-hop neighbor node increases with the increase of overlapping factor. However, the probability of a D-sensor node at same sub region being 1-hop neighbor become low, a main reason is that the key pools of sub regions become larger with the increase of overlapping factor.



(a) (b)

Fig. 6. The probability of being 1-hop neighbor with different *S*

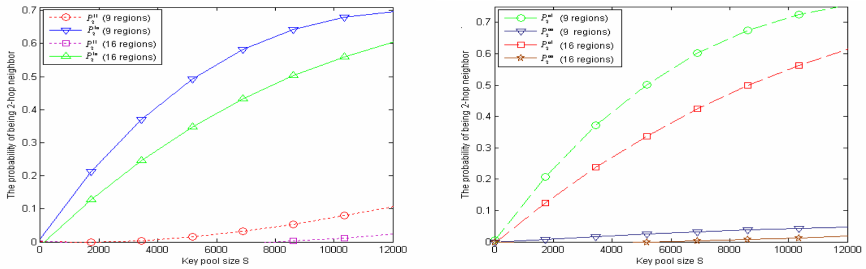


Fig. 7. The probability of being 2-hops neighbor with different *S*

In Fig.7, we also plot the probability of being 2-hops neighbor for different the size of *S* under the overlapping factor *a*=0.13.As shown in Fig.6 (a) and Fig.7, the probability of being 1-hop neighbor node rise with the increasing number of the deployment sub regions. To the contrary, the probability of being 2-hop neighbor node is down. This means that with a high probability a D-sensor node directly shares one key with its cluster head, and this significantly reduces the communication overheads of finding shared key between D-sensors and H-sensors.

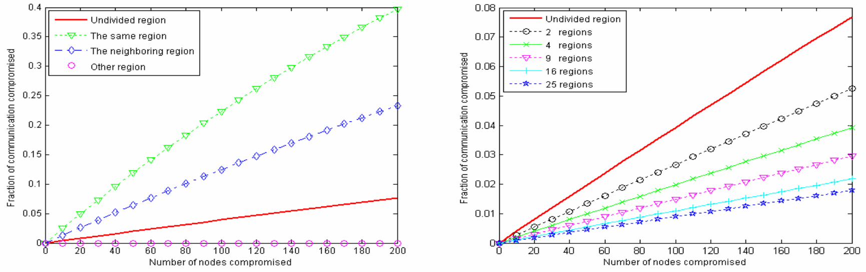
5 Security Analysis

A resilience toward node capture is calculated by estimating the fraction of total network communications that are compromised by a capture of *x* nodes not including the communications in which the compromised nodes are directly involved. To evaluate our scheme against node capture, we apply the same method used in [2,8,10]. In these papers, let the size of key pool *S*, the estimation of the expected fraction of total keys being compromised is calculated by:

$$1-(1-\frac{m}{|S|})^{x} \tag{12}$$

where *x* is the number of compromised nodes, *m* is the size of key ring.

In the previous paper, we suppose that H-sensors have tamper-resistant hardware, so they can not be captured, the security analysis in our scheme only needs to consider D-sensors. We could carry on the security analysis from three aspects: ①the captured nodes are concentrated at the same deployment sub region; ② the captured nodes are randomly deployed at any sensing region. ③compare with existing schemes under the same network connectivity.



(a) (b)

Fig. 8. The network scalability for different the number of compromised nodes

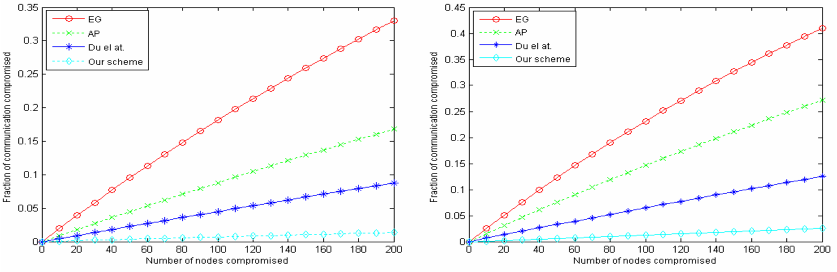
Let *S*=10,000, *a*=0.17, *m*=40, the number of deployment sub regions is 9. In Fig.8(a), we plot the fraction of communications compromised when *x* D-sensor nodes at the same sub region are compromised. As shown in Fig.8 (a), the fraction of compromised communications, which *x* captured D-sensor nodes concentrated at a certain sub region, is higher than the fraction of undivided region. Its neighboring sub regions have lower affection, and other regions are not affected. This indicates that if the captured nodes concentrated at a certain sub region, nodes in this sub region will be badly affected. If an adversary begins from a certain sub region and concentrate at this sub region, he could just extract all keys from compromised nodes at this sub region and its neighboring regions, and monitor communication of this sub region and its neighboring regions. However, he can't control other regions. Fig.8 (b) shows how the fraction of communications compromised varies with the number of compromised nodes at random sub regions. Fig.8 (b) indicates that we can lower the fraction of communications compromised by increase the number of the deployment sub regions.

Since there are two different types of nodes in HWSNs, and the deployment region is divided into many sub regions in our scheme, our scheme doesn't calculate the connective probability P_s by using random graph theory. If we can find another appropriate average connecting probability $\overline{p_s}$, we can let $P_s \geqslant \overline{p_s}$, and achieve a great secure performance and the network connectivity under $\overline{p_s}$. Thus, we can estimate resilience toward node capture in our scheme and compare with the exiting schemes by using $\overline{p_s}$ as:

$$\overline{p_s} = \frac{1}{(1+\lambda)d^2}[\sum_{i=0}^{d} iP(1,m)+(d-i)P(a,m)+\frac{1}{\lambda}\sum_{i=0}^{\lambda d} iP(1,M)+(d-i)P(a,M)] \qquad (13)$$

where d is the degree of D-sensors, $\lambda = d^{'}/d$, $d^{'}$ is the degree of H-sensors.

As shown in Fig.9, we compare our scheme with the existing key management scheme such as Eschenauer *et al.*'s scheme (EG scheme), Du *et al.*'s scheme (AP scheme) and Du *et al.*'s scheme. We can see clearly that our scheme lowers the fraction of compromised communication after x nodes are compromised. The most important reasons for this improvement is that, to achieve the same connectivity while using the same key pool size |*S*|, our scheme only requires much smaller m keys. For instance, to achieve $\overline{p_s} = 0.33$, under |*S*|=100,000, EG scheme [1], AP scheme [10] and Du *et al.*'s scheme [8] respectively require *m*=200, 92 and 46. However, our scheme only needs *m*=7.In the case, the same improvement can be found. Because of the introduction of node deployment knowledge and some powerful nodes, we can effectively reduce the number of keys in the node and save the memory space.



(a) $\overline{p_s} = 0.33$ (b) $\overline{p_s} = 0.50$

Fig. 9. Comparing with existing schemes

6 Conclusions

In this paper, we present a key management in HWSNs, which uses deployment knowledge and prior region information, in order to improve the random key management scheme. The key management scheme shows many advantages by the performance evaluation and security analysis. The probability that two nodes share common key rises with the increasing number of deployment sub regions, which can efficiently reduce communication overhead for finding path-keys. Furthermore, connectivity of neighboring regions could rise with increment of overlapping factor. Simulation results show that the scheme has great scalability, high connectivity, low memory requirement and a stronger resilience toward node capture.

References

1. Carman, D.W., Kruus, P.S.: Matt B.J.Constraints and approaches for distributed sensor security. NAI LABS Technical Report (2000)
2. Eschenauer, L., Gligor, V.: A key management scheme for distributed sensor networks. In: Proceedings of the 9th ACM Conference on Computer and Communication Security, pp. 41–47 (2002)
3. Chan, H., Perrig, A., Song, D.: Random key pre-distribution schemes for sensor networks. In: IEEE Symposium on Research in Security and Privacy, pp. 197–213 (2003)
4. Ren, K., Zeng, K., Lou, W.: A new approach for random key pre-distribution in large-scale wireless sensor networks. Journal of Wireless Communication and Mobile Computing (Special Issue on Wireless Networks Security) 3(6), 307–318 (2006)
5. Du, W., Deng, J., Han, Y.: A pairwise key predistribution scheme for wireless sensor networ-ks. In: Proceedings of the 10th ACM Conference on Computer and Communications Security, pp. 42–51 (2003)
6. Liu, D., Ning, P.: Establishing Pairwise Keys in Distributed Sensor Networks. In: Proceedings of the 10th ACM Conference on Computer and Communications Security, pp. 52–61 (2003)
7. Liu, D., Ning, P.: Location-based pairwise key establishments for static sensor networks. In: Proceedings of SASN 2003, pp. 72–82 (2003)
8. Du, W., Deng, J, Han, Y.: A key management scheme for wireless sensor networks using de-ployment knowledge. In: Proceedings of IEEE INFOCOM 2004, pp. 586–597 (2004)
9. Liu, F., Rivera, M., Cheng, X.: Location-aware Key Establishment in Wireless Sensor Networks. In: IWCMC 2006, pp. 21–26 (2006)
10. Du, X., Xiao, Y., Mohsen, G.: An effective key management scheme for heterogeneous sen-sor networks. Ad Hoc Networks 5(1), 24–34 (2007)
11. Duarte-Melo, E., Liu, M.: Analysis of energy consumption and lifetime of heterogeneous wireless sensor networks. In: Proceedings of IEEE Globecom, Taipei, Taiwan (November 2002)
12. Yarvis, M., Kushalnagar, N., Singh, H., et al.: Exploiting heterogeneity in sensor networks, In: Proceedings of the IEEE INFOCOM 2005, Miami, FL (March 2005)
13. Du, X., Lin, F.: Maintaining differentiated coverage in heterogeneous sensor networks. EURASIP Journal on Wireless Communications and Networking 4, 565–572 (2005)
14. Du, X., Xiao, Y.: Energy Efficient Chessboard Clustering and Routing in Heterogeneous Sensor Network (2005), http://www.cs.ndsu.nodak.edu/~xdu/paper/IJWMC_2005.pdf
15. Lu, Q., Qian, Y., Hu, J.: A framework for distributed key management schemes in heterogeneous wireless sensor networks. In: Proc. of IEEE IPCCC, pp. 513–519 (April 2006)
16. Camtepe, S.A., Yener, B.: Key Distribution Mechanisms for Wireless Sensor Networks: a Survey. Department of Computer Science, Rensselaer Polytechnic Institute, Technical Report TR-05-07 (March 23, 2005)

Author Index